RADIO FREQUENCY SENSING NETWORKS FOR LOCALIZATION, SYNCHRONIZATION, AND HEALTH MONITORING

by

Anh Luong

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

The University of Utah

August 2017

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Anh Luong**
has been approved by the following supervisory committee members:

Neal Patwari ,	Chair(s)	**March 27, 2017** Date Approved
Thomas Schmid ,	Member	**March 27, 2017** Date Approved
Kenneth Stevens ,	Member	**March 27, 2017** Date Approved
Sneha Kasera ,	Member	**March 27, 2017** Date Approved
Ross Walker ,	Member	**March 27, 2017** Date Approved

by **Gianluca Lazzi** , Chair/Dean of
the Department/College/School of **Electrical and Computer Engineering**
and by **David B. Kieda** , Dean of The Graduate School.

ABSTRACT

Low-cost wireless embedded systems can make radio channel measurements for the purposes of radio localization, synchronization, and breathing monitoring. Most of those systems measure the radio channel via the received signal strength indicator (RSSI), which is widely available on inexpensive radio transceivers. However, the use of standard RSSI imposes multiple limitations on the accuracy and reliability of such systems; moreover, higher accuracy is only accessible with very high-cost systems, both in bandwidth and device costs. On the other hand, wireless devices also rely on synchronized notion of time to coordinate tasks (transmit, receive, sleep, etc.), especially in time-based localization systems. Existing solutions use multiple message exchanges to estimate time offset and clock skew, which further increases channel utilization.

In this dissertation, the design of the systems that use RSSI for device-free localization, device-based localization, and breathing monitoring applications are evaluated. Next, the design and evaluation of novel wireless embedded systems are introduced to enable more fine-grained radio signal measurements to the application. I design and study the effect of increasing the resolution of RSSI beyond the typical 1 dB step size, which is the current standard, with a couple of example applications: breathing monitoring and gesture recognition. Lastly, the *Stitch* architecture is then proposed to allow the frequency and time synchronization of multiple nodes' clocks. The prototype platform, Chronos, implements radio frequency synchronization (RFS), which accesses complex baseband samples from a low-power low-cost narrowband radio, estimates the carrier frequency offset, and iteratively drives the difference between two nodes' main local oscillators (LO) to less than 3 parts per billion (ppb). An optimized time synchronization and ranging protocols (EffToF) is designed and implemented to achieve the same timing accuracy as the state-of-the-art but with 59% less utilization of the UWB channel. Based on this dissertation, I could foresee Stitch and RFS further improving the robustness of communications infrastructure to GPS jamming, allow exploration of applications such

as distributed beamforming and MIMO, and enable new highly-synchronous wireless sensing and actuation systems.

For my parents, Xuan Luong and Hien Truong.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude and deepest appreciation to my advisors, Prof. Neal Patwari and Prof. Thomas Schmid for the continuous support during my Ph.D. study and related research, and for all of their patience, motivation, and immense knowledge. Without their guidance and continuous support, this dissertation would not have been possible. I would like to thank both of you for encouraging my research and for allowing me to continue to grow as a researcher. Your advice and positive attitudes towards my research and my career have been absolutely invaluable.

I would like to thank the rest of my Ph.D. committee members, Prof. Ross Walker, Prof. Ken Stevens, and Prof. Sneha Kasera, for their insightful comments, brilliant feedback, and encouragement. I would also like to thank all of you for making my defense an enjoyable moment, and for your constructive criticisms and challenging questions, which motivates me to widen my research from various perspectives.

A special thank you to all of my lab-mates, Dr. Peter Hillyard, Charissa Che, Michael Empey, Andrzej Forys, Kyeong Min, Jon Davies, Enoch Lee, Dr. Maurizio Bocca, Dr. Merrick McCracken, and Alamayehu Solomon Abrar, for their continuous support. My Ph.D. journey has been a great joy and experience thanks to all of you.

Last but not least, I would like to show my deepest gratitude to my family members. Words cannot describe how grateful I am to my mother and my father for all of their sacrifices and support throughout my whole journey of growing and maturing as a person and as a member in the society. I would not have been who I am today without your unconditional love and care. I would also like to thank my mother-in-law for all of her encouragement during my journey and for trusting me with her beloved daughter. At the end, I would like to express my love and appreciation to my dearest wife, Anh Phan, who loves me with all her heart and who is always be there for me to be my support even in my darkest moments. Thank you for always being on my side, for being yourself, and for being my best friend.

CHAPTER 1

INTRODUCTION

1.1 Motivating Example

On April 24th, 2013, the Savar building in Bangladesh collapsed and buried 3,122 people underneath; however, only 2,515 lucky survivors were rescued from the rubble while 1,129 unfortunately did not survive. It took the search and rescue team more than 17 days to finish searching and they still found a survivor on the last day [1]. Time to rescue those trapped in emergency situations for search and rescue teams is extremely limited; the faster they are able to find and extricate trapped people, the more lives they can save. Rescuing efforts may have been expedited if there were sensors that could quickly be deployed (thrown or placed) around the search area to locate people and estimate their condition. Of the four vital signs (body temperature, pulse rate, respiratory rate, and blood pressure), breathing is often used as a first sign to determine whether a person is alive or not. With that in mind, a so-called breathing detection and person localization system could help the search and rescue personnel to quickly determine their priorities to save as many surviving victims as possible.

The goal of this dissertation is to develop, advance, and explore technologies that allow radio devices to be used for localization, synchronization, and health monitoring. Localization takes multiple forms based on the application. As an example, localization could mean "device localization", that is, locating radio devices with respect to each other. In other words, the deployed radio sensors can self-localize in the above search and rescue application to dramatically shorten setup time. In an outdoor-only environment, global positioning system (GPS) would be a reasonable approach to the device localization problem. Differential GPS can achieve 15cm accuracy [2]. However, indoor environments are much different. GPS signals are intermittent and the receivers are energy intensive. In indoor localization, we approach the problem quite differently. Often, commercial indoor

localization algorithms utilize measurements from existing WiFi networks, which is also energy intensive and requires a dense deployed WiFi network. In another approach, we could use the radio devices that can be worn by individuals to measure their proximity to other people. By contact sensing, we could also measure possible exposure to human-borne pathogens, which is a form of health monitoring.

In many situations, the assumption that the subject of interest would always wear a sensor is out of the question. For instance, when rescuers are trying to locate people in a collapsed building, or the police are trying to locate people in a building during a hostage situation, we cannot expect people will always be carrying a wireless device and using it in a way that can be located. Another form of localization is the localization of people who cannot be assumed to be wearing a radio device, referred to as "device-free localization" (DFL) [3]. This method of localization takes advantage of the fact that human motion and activity in the vicinity of a wireless link causes temporal fading to the received signal on that link. A significant portion of DFL research uses devices that can measure received signal strength (RSS), as such measurements are common on inexpensive transceivers. Recently, noninvasive breathing monitoring [4] is another device-free health monitoring technology, as it does not require the person being monitored to wear a device.

Wireless devices have, until recently, been a data exchange medium, and as such, they provide channel measurements only for the purpose of improving the performance of the communication system. Today, received signal strength indicator (RSSI) is being used by researchers as the default measurement for monitoring the state of the environment. Those concerned with the limitations of RSSI have resorted to wideband measurements, WiFi channel state information (CSI), or ultra wideband impulse response (UWB-IR). WiFi CSI and UWB-IR provide more measurements and are more reliable than RSSI. However, for radio frequency (RF) monitoring to become ubiquitous, they must also be bandwidth efficient. In fact, we cannot use 20 MHz at a 2500 packets per second rate, as proposed by [5], purely for one RF activity monitor. Moreover, time-based localization has recently gained significant importance thanks to the commercial development of low-cost UWB-IR transceivers. DecaWave has developed low-cost RFICs (10 USD) that can achieve 39 cm range accuracy in an indoor environment [6]. However, it achieves such a feat by using 500 to 1000 MHz of bandwidth, and requires networks to transmit nearly continuously

on this channel to obtain network synchronization. The limitation indeed relies on the number of devices that could be deployed simultaneously due to bandwidth limitations. DecaWave does its standard ranging by measuring the time of flight, which requires the devices to agree on epoch and duration between the epochs. Each device contains a clock. A clock is a timer that counts using a predefined period that is often provided by an oscillator. This oscillator is local to each device and often called a local oscillator. Each device tends to contain different local oscillators. Depending on the types, its characteristics and its circuitry, the period could be varied. Thus, in order for devices to agree on the time, we often need upward of six messages per every pair of nodes to estimate the main local oscillator characteristics (offset and skew).

1.2 Contributions

In essence, my dissertation explores and expands the boundaries of the state of the art in low-cost, low-bandwidth RF sensing. The methods and systems described in this dissertation will provide fundamentally new sensing capabilities that are well-suited for large-scale commercial adoption and in "smart" devices and systems.

First, I address my work using the standard RSSI provided by off-the-shelf radio transceivers for purposes of localization, synchronization, and health monitoring. Many applications are limited when only RSSI can be used. Current systems that can obtain more accurate radio channel information are often expensive. As a result, the methods that can be used in RF embedded systems are discussed and introduced in this paper to achieve both low cost and low bandwidth, and yet improve the ability of RF devices to perform in localization, synchronization, and health monitoring applications.

1.2.1 Use of Standard RSSI

Chapter 2 introduces WRENSys to address a specific need for the measurement of contact between people in the monitoring of exposure and spread of pathogens. The work was motivated by a desire from epidemiologists to have a system that could be deployed to study contact among school-aged children. Their application had particular constraints and required features. First, the system must allow hundreds of children to have contact sensors. At the start of each school day, the sensors are deployed to students

and collected afterwards; thus, the deployment and collection time must be minimal per sensor. Second, the system must accurately detect and log any other contact sensor within a 1-3m radius, which effectively allows researchers to pinpoint which participants were within a disease's contagion distance. Due to the sheer number of devices, the system had to have low cost and maintenance time, including the time to charge and download contact data from the devices. In this joint work, I designed and implemented the subsystem for built-in large-scale mass programming, including hardware design, firmware, and software. The following publications have resulted from this work.

1. Andrzej Forys, Anh Luong, Enoch Lee, Jon Davies, Kyeong Min, and Thomas Schmid, "Rapid Deployable System for Human Contact Network Research", Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems, 2012 [7].

    - Beside the previously stated contribution, I designed and implemented a contact sensing application. This application tracks contacts between each tags, WRENs carried by patrons, and anchors, WRENs placed at each demo/poster booth. This demonstration has received "SenSys'12 Best Demo Award".

2. Kyeong T. Min, Andrzej Forys, Anh Luong, Enoch Lee, Jon Davies, and Thomas Schmid, "WRENSys: Large-scale, Rapid Deployable Mobile Sensing System", 39th Annual IEEE Conference on Local Computer Networks Workshops, 2012 [8].

    - On top of designing a mass programming subsystem, I was part of designing the WREN system, and deployment and data collecting logistics for other systems involved in this project.

Chapter 3 presents the Dial-It-In system. The performance of RSS-based DFL, as we demonstrate in [9], is highly dependent on the multipath fading experienced between the static deployed transceivers. Due to small-scale fading, the exact position of the antenna determines whether the multipath components add in a destructive manner and the link is in a "deep fade," or if they add in a constructive manner and put the link in an "anti-fade." In Dial-It-In, we created a system that automatically moved sensors in order to optimize the fading condition of the channels in the network to improve localization performance. The following publication has resulted from this work.

1. Maurizio Bocca, Anh Luong, Neal Patwari, and Thomas Schmid, "Dial It In: Rotating RF Sensors to Enhance Radio Tomography", Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 2014 [9].

    - In this joint work, I designed and implemented the hardware and firmware to automatically moved sensors in order to optimize the fading condition of the channels. I was also a part of deployment and testing of the full localization system.

Weather conditions (rain, snow, wind, etc.) could easily alter and increase the variability of the RSS on a channel as shown in Chapter 4. Therefore, an outdoor RSS-based DFL system that could reliably predict and localize boundary crossings despite changes in weather was developed and deployed over months. The following publication has resulted from this work.

1. Peter Hillyard, Anh Luong, and Neal Patwari, "Highly Reliable Signal Strength-based Boundary Crossing Localization in Outdoor Time-Varying Environments", Proceedings of the 15th International Conference on Information Processing in Sensor Networks (IPSN), 2016 [10].

    - In this joint work, I was a part of designing and implementing a system for boundary crossing detection using the TI CC2531 and a custom energy harvesting solution for long-term outdoor deployment.

Chapter 5 describes a real-time breathing monitoring system that uses two transceivers, both measuring the RSSI of received packets, to estimate breathing rate [11]. In comparison, existing state-of-the-art prototype systems were non-real-time systems [4], [12]. The following publication has resulted from this work.

1. Anh Luong, Spencer Madsen, Michael Empey, and Neal Patwari, "RUBreathing: Non-Contact Real Time Respiratory Rate Monitoring System", Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN), 2015 [11].

- I designed and implemented a real-time breathing monitoring system with the CC2530 nodes that were able to explore the limitations that then inspired me to create the sub-dB system.

1.2.2 Novel Channel Measurement Systems

1.2.2.1 sub-dB

In addition, Chapter 6 will also describe a new narrowband RF measurement technology, sub-dB. Sub-dB provides higher accuracy received signal strength (RSS) measurement. I believe the main reason that the RSSI from standard radios is inadequate for many sensing applications is its step size. A 1 dB RSSI step size, the smallest step size that is commonly available, is not capable of relaying small changes in the channel. As a result, the design and evaluation of a system that uses a TI CC1200 transceiver and a Beaglebone processor to make measurements of RSS with 0.01 dB median error is presented for that purpose. The result proves that the sub-dB system outperforms the previous RSSI and multichannel RSSI methods [13]. Moreoever, it uses much less bandwidth than any other systems. Thus, it could be integrated seamlessly into existing communication systems. The following publication has resulted from this work.

1. Anh Luong, Alameyehu Solomon Abrar, Thomas Schmid, and Neal Patwari, "RSS Step Size: 1 dB Is Not Enough!", Proceedings of the 3rd Workshop on Hot Topics in Wireless (HotWireless), 2016 [13].

    - In this work, I built a sub-dB system prototype and evaluated the performance with a breathing monitoring application.

1.2.2.2 Frequency and Time Synchronization

Chapter 7 introduces the system for bandwidth efficient frequency and time synchronization. Prior to this work, a wireless large-scale system that requires high level of synchronization was very challenging (i.e., distributed MIMO, wireless beamforming, large-scale cm-level localization, etc...). This particular work addresses these challenges in the most fundamental way, efficiency of clock synchronization. The TI CC1200 would expose IQ (complex baseband) samples immediately after the CORDIC algorithm. I will show the ability to make use of the signal phase measurements for frequency synchronization.

The Chronos platform implements the Stitch architecture, which could also be used as a playground for a wide range of clock-related research and development including the unique ability to change and adapt the clock network to tailored applications. Normally, this is expensive to implement because it requires us to build separate platforms, thus increasing development and fabrication costs and time. Furthermore, even though the technical contributions show the use of a single set of hardware, the idea behind Stitch could be applied on other sets of hardware. The following publications have resulted from this work.

1. Anh Luong, Thomas Schmid, and Neal Patwari, "A Platform Enabling Local Oscillator Frequency Synchronization", Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems (SenSys), 2015 [14].

    - I designed and built a platform to explore wireless frequency synchronization through low-cost narrowband radio. The system was later demonstrated at a conference and received much invaluable feedback.

2. Anh Luong, Peter Hillyard, Alameyehu Solomon Abrar, Charissa Che, Thomas Schmid, and Neal Patwari, "A Stitch in Time and Frequency Synchronization Saves Bandwidth", unpublished.

    - In this joint work, I designed and introduced the Chronos platform based on the proposed Stitch architecture. I also implemented and evaluated the RFS algorithm for frequency synchronization of two independent oscillators. With that, I was part of the designing and implementing of EffToF, a protocol that efficiently estimates time-of-flight measurement for synchronization or ranging. Stitch, Chronos, RFS, and EffToF are all discussed in Chapter 5.

Finally, Chapter 8 summarizes research findings in this dissertation as well as opportunities for future research.

1.3 Copyright Notice

1.4 References

[1] Jim Yardley, "Report on deadly factory collapse in bangladesh finds widespread blame," 2013.

[2] Dmitry Kozlov and Michael Tkachenko, "Centimeter-level, real-time kinematic positioning with gps+glonass c/a receivers," *Navigation*, vol. 45, no. 2, pp. 137–147, 1998.

[3] Maurizio Bocca, Ossi Kaltiokallio, and Neal Patwari, *Radio tomographic imaging for ambient assisted living*, pp. 108–130, Springer, 2013.

[4] Ossi Johannes Kaltiokallio, Hüseyin Yigitler, Riku Jäntti, and Neal Patwari, "Non-invasive respiration rate monitoring using a single COTS TX-RX pair," in *ACM/IEEE IPSN*, 2014, pp. 59–70.

[5] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *ACM MobiCom*, 2015, pp. 65–76.

[6] Dimitrios Lymberopoulos, Romit Roy Choudhury, Jie Liu, Souvik Sen, Xue Yang, and Vlado Handzinski, "Microsoft indoor localization competition: Experiences and lessons learned," *SIGMOBILE Mobile Computation and Communication Review (MC2R)*, October 2014.

[7] Andrzej Forys et al., "Rapid deployable system for human contact network research," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 383–384.

[8] Kyeong T Min, Andrzej Forys, Anh Luong, Enoch Lee, Jon Davies, and Thomas Schmid, "Wrensys: Large-scale, rapid deployable mobile sensing system," in *Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on*. IEEE, 2014, pp. 557–565.

[9] Maurizio Bocca, Anh Luong, Neal Patwari, and Thomas Schmid, "Dial it in: Rotating rf sensors to enhance radio tomography," in *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2014, pp. 600–608.

[10] Peter Hillyard, Anh Luong, and Neal Patwari, "Highly reliable signal strength-based boundary crossing localization in outdoor time-varying environments," in *ACM/IEEE IPSN*, April 2016, pp. 1–12.

[11] Anh Luong, Spencer Madsen, Michael Empey, and Neal Patwari, "Rubreathing: Non-contact real time respiratory rate monitoring system," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, New York, NY, USA, 2015, IPSN '15, pp. 412–413, ACM.

[12] Neal Patwari, Lara Brewer, Quinn Tate, Ossi Kaltiokallio, and Maurizio Bocca, "Breathfinding: A wireless network that monitors and locates breathing in a home," *IEEE J. Sel. Topics in Signal Processing*, pp. 30–42, Feb. 2014.

[13] Anh Luong, Alemayehu Solomon Abrar, Thomas Schmid, and Neal Patwari, "Rss step size: 1 db is not enough!," in *Proceedings of the 3rd Workshop on Hot Topics in Wireless*. ACM, 2016, pp. 17–21.

[14] Anh Luong, Thomas Schmid, and Neal Patwari, "A platform enabling local oscillator frequency synchronization: Demo abstract," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 300–301.

PART I

DEVICE-BASED LOCALIZATION

CHAPTER 2

WRENSYS: LARGE-SCALE, RAPID DEPLOYABLE MOBILE SENSING SYSTEM

2.1 Abstract

We present *WRENSys*, a system that allows for a low-cost, rapidly deployable, large-scale and easy-to-maintain wireless mobile network, with minimal subject interference, to study contact networks of a population. We discuss our experiences with several hardware designs, including our new *WREN* sensor, base stations for charging and programming 2000 motes in parallel, software applications, and our deployment experiences and results. We deployed the system at 26 different locations with an average of 500 participants per school, the largest having more than 1,500 sensors, resulting in 35 million contact data points. On average, our system required only 30 minutes of preparation time and 30 minutes of deployment time with one graduate student and two additional personnel. Our system allowed for contact networks with coverages at approximately 80%. This shows a sustainable platform for studying human contact networks. [1]

2.2 Introduction

The influenza virus affects hundreds of children and elderly people yearly in a deadly manner. While over 100 children died from the flu in 2013, during the epidemic of 2009-2010, about 350 children lost their lives due to the flu [1]. Experts believe the flu virus spreads mainly by expelled droplets when infected individuals cough, sneeze, or talk.

[1] In this joint work, I designed and implemented the subsystem for built-in large-scale mass programming, including hardware design, firmware and software. On top of mass programming subsystem, I was part of designing the WREN system, and deployment and data collecting logistics for other systems involved in the project. This chapter appears in 39th Annual IEEE Conference on Local Computer Networks Workshops, "WRENSys: Large-scale, rapid deployable mobile sensing system", Kyeong T. Min, Andrzej Forys, Anh Luong, Enoch Lee, Jon Davies, and Thomas Schmid.

In "How Flu Spreads" [2], the Centers for Disease Control and Prevention (CDC) of the United States reports the flu can potentially infect others at a distance of up to 2 meters. Therefore, we focus on studying the human contact network of school-aged children, how long they are in proximity with whom, and how these contacts mix between the different classes and grade levels during recess, lunch breaks, extracurricular activities, and homes. Measuring human contact networks by equipping people with contact sensors allows researchers to understand and study how airborne diseases spread by using real-life data sets instead of relying on computer population models. Also, the capabilities of large-scale, rapidly deployable systems enable researchers to include larger populations and gain statistically meaningful data and models. Figure 2.1 shows a human contact network graph depicting the complex data and contact patterns between students throughout a school day with *WRENSys*.

Contact among Utah School-age Population (*CUSP*) aims to extract real-life human contact networks in schools using wireless sensors for the purpose of modeling the transmission of airborne diseases. Performing this study is challenging, however, as every study subject has to be equipped with a sensor. Our deployments did not only include high schools, where we could expect large concentrations of smartphones, but also middle and elementary schools, as well as kindergartens, in all of which smartphone usage could be prohibited. In addition to the large age-span of study subjects, our plan was to deploy the system at different schools on a weekly basis with minimal interference to daily school activities. Therefore, the system must be prepared, configured, programmed, and deployed within a 30-minute window while scaling to 2000 sensors. This leads to special requirements and design approaches for various challenges such as sensor maintenance, system-wide programming, data collection, and decreases in unit cost for large-scale rapidly deployable sensing systems.

The primary requirements for the *CUSP* project are:

- accurately detect contacts within a 1-3 meter range
- enable high density coverage ($\sim$2,000 sensors)
- unobtrusive, small, comfortable, and safe to wear
- easy maintenance, charging, faulty node detection

- parallel programming and data downloading
- repeatable, rapid deployments
- minimal development schedule

2.3 Background

The advent of smartphones and their large penetration in society make these devices the ideal tool to study large-scale human populations for data such as: localization, movement patters, or to determine what places are "hot" right now [3]. Although smartphones have been gaining popularity in researching human contact studies, how do we deal with the heterogeneity inherent in smartphone hardware, reliable deployments, application upkeep, and various types of users?

In [4], Ahmad et al. studied various usage models of the iPhone and its dependence on the socioeconomic status (SES) of a person. They find that different SES groups use the iPhone very diversely, and suggest various design implications to support studies across SES levels. Furthermore, Nathan et al. indicate in [5] that while the Reality Mining study supplied 100 identical mobile phones as wearable sensors, their localization was limited to an accuracy of 10 meters. This would be unusable in human contact studies for influenza that require the 2-meter range recommended by the CDC. Furthermore, not all students, especially kindergartners, have smartphones and policies can limit access to phones during the day.

While wireless system with similar goals have been deployed before in schools [6], it became evident to us after experimenting and deploying commercially available nodes that they do not meet our requirements. We found severe drawbacks and limitations with those solutions, such as high cost, nonscalability, and maintainability issues, which motivated the design of a new platform. For example, the wireless nodes used in [6], the TelosB, are powered by two AA batteries and lack a recharging mechanism that does not involve replacing all the batteries. Thus, after our initial deployments, we began developing our own platform while substituting the TelosB nodes with Irenes, which unfortunately presented other drawbacks discussed later.

WRENSys addresses the aforementioned issues by re-engineering the whole Wireless Sensor Network (WSN) system stack, from hardware including wireless nodes (sensing,

physical design, weight, maintenance, battery life; see Figure 2.2) and base stations (maintenance, recharging, reprogramming) to supporting applications. We show our results of *WRENSys* and discuss the comparison to existing hardware platforms such as the TelosB and Irene in Section 2.8. With *WRENSys*, one graduate student can provision and manage up to 2000 nodes, all within 30 minutes, truly allowing for rapid, successful deployments.

Our previously published work on this topic includes a demo [7] and data mining workshop [8]. We demonstrated our proof-of-concept hardware design at [7] to show a functional system using the WREN nodes. Since then, we have completed the majority of our deployments and experienced new insights to sustain such a system. While a brief description of *WRENSys* occurs in [8] to give the reader an understanding of the platform, that paper focuses on actual deployment challenges, data collection and processing, and network validation. Given the amount of raw data, we needed to manage it correctly during and post deployment, and develop methods for better node synchronization as detailed in [8].

Our key contributions are, first of all, to meet all aforementioned system requirements and to provide a design and implementation of a rapid, large-scale, low-maintenance wireless sensing system with design approaches that set the groundwork for further research. Future applications are not limited to social contact network study, but can include other applications where the goal of rapid and large-scale deployment is paramount to the system requirements. Importantly, the scope of this paper does not delve into the epidemiological contact data analysis, which is planned to be analyzed by *CUSP*. All hardware schematics, base station construction, TinyOS *WREN* platform, and application software code are open source and publicly available at [9].

2.4 Related Work

In [6], Kazandjieva et al. used 994 TelosB sensors inside a conference badge with a lanyard for a one-day, one-time deployment in a high school in California to measure its social contact network. Deploying a large number of TelosB sensors requires considerable effort. Kazandjieva reported issues with hardware resets, battery disconnects, and uncomfortable lanyards that would swing around the neck of the study subjects. Yet the collected contact data were extremely insightful, and were published in [10]. As

we will describe later, we attempted using TelosB nodes, but found that even 200 nodes are impractical for a rapidly deployable low-maintenance system. *WREN* nodes utilize a light-weight, rechargeable Li-Ion Battery and are designed to fit in a small, two-part plastic enclosure.

The OpenBeacon Project [11] developed a low-cost active RFID tag for similar contact studies as with the *WRENSys* targets. The main difference between the OpenBeacon tag and our developed *WREN* is the required infrastructure of the OpenBeacon. Having this infrastructure simplifies the tags significantly, since no data must be stored locally on the node. The OpenBeacon tag measures the RSS of an incoming message, then switches to the infrastructure channel and sends a message to a server over the installed wireless base stations, recording this information. Installing the infrastructure is possible, for example, during a conference where Ethernet plugs and Internet connectivity is readily available. However, our targeted schools do not have such an infrastructure, and deploying the wireless base stations adds significant deployment overhead. Thus, we decided to include additional storage capabilities on our *WREN* nodes, such that any infrastructure at deployment sites is unnecessary.

OpenBeacon's sister project, an EU-funded project called SocioPatterns [12], used a similar system in a primary school to gather their high-resolution contact network [13]. The system was deployed for 2 days and collected 77,602 contacts between 242 individuals. In contrast, *WRENSys* has now been deployed in over 26 different locations, averaging 2 days deployments, and collected over 35 million contacts so far.

2.5 Commercial Platforms

We developed three major hardware iterations (Figure 2.2), learning from every design to improve the next. The original plan was to use commercially available wireless sensor nodes in order to quickly start studying contact networks. Our positive and negative experiences throughout each iteration of sensors, base stations, and software influenced our subsequent platform design decisions.

2.5.1 TelosB Platform

The first deployment involved a network of 200 TelosB nodes (Figure 2.2, Figure 2.3). While these nodes are excellent for small networks, they lack an efficient and scalable mass programming option. This limit is imposed by the nature of the USB host communicating with UART bootloaders on the TelosB nodes. The TelosB nodes also require new batteries after a few days of deployment. Simply replacing the batteries required several hours of work by over five people to prepare for each deployment.

We designed a case to hold the nodes and protect them from the adolescents who carried them (see Figure 2.2). After multiple enclosure iterations, we decided on a custom, laser-cut acrylic design reinforced with packing tape. While the case avoided unexpected resets due to button pushes as reported in [6], it made battery replacement even more tedious as the case had to be taken apart to access the batteries. We also noticed the battery holder and heavy AA batteries allowed for a battery disconnect when the case was shaken during bumps or running activities. This introduced severe problems with time synchronization and algorithms, necessitating heavy data postprocessing to fix time discontinuities.

The TelosB uses USB for programming; thus, we developed four 50-port USB base stations (Figure 2.3) allowing us to connect all 200 nodes to one computer using two independent USB buses. We found that our Linux operating system (Ubuntu 12.04) was not able to enumerate all the connected devices, forcing us to use two laptops to program the whole network of 200 nodes. While we considered wireless programming, the amount of debugging required for this type of system when programming failures occur would be too much for the planned crew of nontechnical deployment personnel. Thus, we abandoned the idea of scaling this system up to the required 2,000 nodes, as it would have been unmanageable [14].

2.5.2 Irene Platform

Due to the limitations of TelosB platform, our second network deployment comprised 200 Irene nodes (Figure 2.2, Figure 2.4). The Irene node is based on the Epic hardware module [15], and contains the same microcontroller and radio chip as the TelosB nodes. Different from the TelosB, Irenes come with a rechargeable battery and small, light-weight

case, which greatly reduced our maintenance time and increased wearable comfort. Unfortunately, the same USB programming limit experienced with the TelosB nodes plagues the Irenes. The battery charging circuit header needed to be epoxied on every node due to breakage after repeatable cable insertions. The Irene nodes also cost significantly more, approximately $200 each, which prohibits minimal budgets from scaling to 2,000 nodes. Although several TelosB drawbacks were circumvented, those encountered using the Irene platform were enough to influence our decision to build a new node from the ground up, which is explained in detail in Section 2.6.

2.5.3 TelosB and Irene Base Station

We built four base stations for the TelosB and Irene platforms. They were heavy (8.4 kg each, without nodes), bulky, and contain a clutter of 50 USB cables for up to 50 nodes per base station, as shown in Figure 2.3. During the transition from TelosB to Irene nodes, we rerouted and replaced each USB Type A cable for the corresponding USB Mini Type B cable used with the Irenes.

Finding nodes that failed to program was a challenge due to the generally random orientation of nodes. The primary drawback was the inability to program over 150 nodes from our laptop using USB. Although deployments succeeded, sensors took over one hour to distribute and needed a substantial labor force to complete. These interface limitations proved unsustainable in deployments of several hundred nodes. We developed a new type of base station to accompany our new *WREN* nodes that improved upon the initial drawbacks and allowed for a consistently rapid-deployable system.

2.5.4 Initial Enclosure

We carefully considered the enclosure for the TelosB and Irene nodes for safety, low-cost, sensor protection, and user-friendliness/wearability for study subjects. Figures 2.2 and 2.4 show the clear acrylic TelosB and Irene cases, respectively. We ground down the sharp corners and covered the edges with clear tape for the students' safety when using the TelosB cases. To avoid breaking the case, lanyards were attached through the battery connector on the node as stress relief. We found this case unfit for continuous deployment due to the the high maintenance and the size and weight of the entire unit. Less bulky

cases are provided with Irene nodes, but instead of using lanyards, we decided to make custom Velcro bands such that the students wear them on their wrists. However, this led to problems such as accounting for variable wrist sizes and bands easily becoming dirty or blocking the charging port. While our previous enclosures were carefully considered and used for several deployments, the case design for our *WREN* sensors minimized all the drawbacks we experienced using the TelosB and Irene case designs.

2.6 WREN Platform

In this section, we describe our new platform, architecture, and details of *WRENSys*. The *WREN* hardware architecture consists of our node, base station, and custom boot strap loader (BSL) based on the I2C serial bus.

2.6.1 Wireless Ranging Enabled Node (WREN)

An important design consideration is to decide whether to adhere to the same hardware families as the TelosB and Irene, or to switch to a new architecture. The advantage of staying is an easier port of already existing software, such as the software applications explained in Section 2.7 designed for TelosB and Irene nodes. However, the hardware used on the TelosB is almost 10 years old, and significant improvements have been made in microcontroller and radio designs since then. Therefore, we decided to redo the design from the ground up, while keeping some software compatibility, as the advantages of cost cutting choices, feature improvements, and lower power draws outweighed the disadvantages.

2.6.1.1 Microcontroller

The entire node design is based around the choice of the microcontroller. The venerable TI MSP430F1611 is the heart of both the TelosB and Irene nodes. This microcontroller, while well supported, has a low feature-to-cost ratio ($10 @1K units). Since price was an important metric, we considered a more modern and cost-optimized microcontroller. Table 2.1 compares the three platforms. We chose the TI MSP430F5342 ($3 @1K units, Figure 2.5) for several reasons:

- *Code compatibility*: The MSP430 5 series run the same Instruction Set Architecture (ISA) as the 1 series. This allows a swift transition from the old software develop-

ment platform.

- *Higher clock frequency*: With a top clock frequency of 25 MHz versus 8 MHz, in the event of more processor-intensive tasks, the processing power is available.

- *Lower power consumption*: The CPU core in the 5 series draws 290 μA/MHz at 3 V while the 1 series draws 500 μA/MHz at 3 V.

- *High storage capacity*: 10k of RAM and 128k of ROM provides enough storage for even complex applications.

2.6.1.2 Radio

We chose the Atmel AT86RF233 because of its IEEE 802.15.4 compliant radio as well as several key features that significantly improve over the older TI CC2420 used in the Irene and TelosB. We use the measure of radio signal strength indicators (RSSI) between nodes to determine the distance. Our unique design employs an antenna fabricated within the PCB operating at 2.4 GHz, which reduces cost and component space.

- *Transmit power and receiver sensitivity*: The AT86RF233 has a maximum transmit power of +4 dBm (vs. 0 dBm), and a receiver sensitivity of -101 dBm (vs. -94 dBm). This increase in link budget of 11 dB allows a considerably longer range than the CC2420. Our own measurements of our final node show a range in excess of 100 meters on an open field at full power.

- *Lower power draw*: In addition to the overall power reduction afforded by continuous operation in both receive and transmit (see Table 2.2), receive power consumption can be further reduced by up to 50 percent using Smart Receiving Techniques implemented in receive mode.

- *Higher data rate*: Bypassing the chip mode of the IEEE 802.15.4 radio, the AT86RF233 achieves data rates as high as 2 Mbps. One power-saving technique employed in the integrated circuit (IC) is to desensitize the receiver. This reduces power, at the cost of not detecting messages with a very low RSSI. Desensitizing the receiver was necessary in order to reduce the maximum receiver range to the required 2-3 meters

at the lowest transmit power, thus reducing the total number of overheard messages from a larger neighborhood. Figure 2.6 shows the RSSI given several range tests.

Another reason for choosing the AT86RF233 was its potential for RF ranging. It integrates a phase measurement unit, giving the possibility of a phase-based ranging systems. However, this capability has not been explored in the current software architecture.

2.6.1.3 Power system

The power system employed on the *WREN* revolves around the TI BQ25010, a lithium ion/polymer charging IC that includes a high-efficiency switching regulator that is used as the primary voltage source for the system. This regulator provides power duty cycling that enables high-efficiency at a wide range of system loads by changing the switching frequency depending on power demand. For example, during system sleep, the power supply reduces its switching frequency to reduce the quiescent current. On average, the *WREN* draws 5.2 mW while executing our application (see Figure 2.7). During deep sleep, it draws 60 μW, of which 50 μW accounts for the power supply's quiescent power. Radio transmit draws 42 mW. These power saving features can sustain deployments for up to 1 work week on a single, 180 mAh battery included in the *WREN*.

2.6.2 WREN Base Station

We designed a base station capable of supporting large-scale sensor networks with solutions for charging, programming, debugging, and rapid-deployment in mind.

We considered the following design requirements for the base station.

- simple interface for power and programming
- charge/program of at least 100 nodes each
- easy access to connect/monitor nodes
- contain status indicators for debugging
- modularity for construction and repair
- minimal size and weight

2.6.2.1 Implementation

The main concept for the base station was to eliminate the bulky USB cabling. We designed a PCB backbone allowing each node to vertically plug into the base station. To ensure proper connections and handling, a sturdy 10-pin Hirose connector was used that is rated for up to 10,000 insertions. We designed PCBs to be modular for construction and debugging purposes. Each PCB module, which we call a ”blade,” supports up to 10 nodes and routes signals from the previous blade to the next (see Figure 2.8). A single base station has 10 interconnected blades, supporting 100 nodes. The programming interface was updated (see Section 2.6.3) to support programming multiple base stations from a single bus. Debugging LEDs on each blade show power (5 V and 3.3 V) and reset status of the nodes. See Figure 2.8 for a base station with a clear, acrylic support structure and blade design.

We designed the support structure for the PCBs and nodes using layers of thin, light-weight laser cut wood. Each base station can be stacked on top of another to decrease size (see Figure 2.9). Even when fully populated, our new base stations only weigh 5.6 kg, which is significantly less than even the unpopulated base stations used for the TelosB and Irene nodes.

2.6.2.2 Power testing

We monitored power usage to determine how long it takes a fully populated base station to finish charging when preparing for deployments. As seen in Figure 2.10, all 100 discharged nodes are fully charged within 2.5 hrs. Once charged, the base station draws approximately 249 mA, which accounts for 100 nodes programmed to sleep and a status LED on each node indicating full charge.

2.6.3 Inter-Integrated Circuit (I2C) Parallel Programming

WRENSys requires support from robust programming to minimize node failure during deployment. It supports fast, reliable, and parallel programming of thousands of nodes. We designed the programming interface around a more scalable shared I2C bus versus USB due to the maximum USB device shortcomings. We briefly considered a wireless programming system, but rejected it due to the large possibility of programming

failures, and the existing requirement of a physical charging station.

2.6.3.1 Bootstrap loader (BSL) design

The BSL on the MSP430F1611 was fixed in ROM, and changing it was impossible. The new MSP430F5342 contains a special flash region for the BSL, allowing us to rewrite its communication logic. We wrote a custom BSL for the *WREN* based on a shared I2C programming interface. The *WREN* enters the BSL only if it detects an active I2C connection (SCL and SDA pulled high) after a hardware reset. This prevents erroneous reboots from getting stuck in the boot sequence.

The MSP430's I2C accepts general broadcast instead of just single point-to-point communication. This allows simultaneous sending of programming packets to all the nodes. The response packets from the nodes are queued individually along with their unique addresses, providing a constant time scaling on the programming bus as we increase the number of nodes.

2.6.3.2 Addressing

I2C requires a 7 or 10 bit address for master-slave communication. Generating a per-base unique address can be challenging. The MSP430 provides an internal unique 32-bit chip ID, but hashing this address down to 7 bits is prone to many address collisions due to the large number of total nodes. Our scalable approach creates a large voltage divider on the base station, which is used in conjunction with the MSP430's Analog to Digital Converter (ADC) to create 100 distinct addresses for the I2C bus protocol. The voltage divider contains series resistors, one per node, powered by 3.3 V. Each node reads the voltage at its resistor using the ADC with 12-bits resolution. The 7 most significant bits will be the node's unique I2C address. This addressing scheme allows communication between the programmer and individual nodes for programming, testing, and debugging. This design gives us a flexible system where placement of the node on the base station does not affect the programming protocol.

2.6.4 *WREN* Enclosure

We approached the *WREN* enclosure selection differently from those of the TelosB and Irene nodes because we were able to design the hardware for the case. The following

are the major design considerations that we took into account: minimal size and weight, debug light, easy connection to base station, and above all: *safety*. We use the Polycase AB-30 series case with minor modifications. This selection reduced the size and weight of the *WREN* node. Figure 2.2 shows our final *WREN* case that fits our specifications. We added a passive RFID chip for node identification in case they don't reply to regular RF communication, and to give deployment staff an option to quickly query the node using a NFC-equipped phone.

2.7 Software Applications

To meet the requirements for *WRENSys*, we used Python for user interface applications, and TinyOS and nesC to program all sensor applications such that device drivers for each TI MSP430 processor for three hardware iterations would be compatible. The supporting software applications are mainly grouped into two categories: stationary and mobile applications. The stationary applications are placed at the deployment center and executed to prepare for and to collect data after deployments. Specifically, they consist of a User Commander, a Command Controller, and a Wireless Downloader. The mobile applications include a Mote Client and a Mote Sleeper application.

2.7.1 Stationary Applications

2.7.1.1 User commander

For the front-end user interaction with sensors, we developed the user commander application (UCA) in Python, which allows actions such as start/stop sensing, get status, blink LEDs, run test routines, start wireless download, erase flash, and find certain nodes. The UCA provides a command prompt interface for accepting commands from the user and forwarding them to the command controller application.

2.7.1.2 Command controller

The command controller application (CCA) is installed on a single commanding node, a TelosB node, which in turn broadcasts commands to client nodes using a predefined radio channel. The reason for using a TelosB instead of a WREN is for the convenience of a direct USB interface for the wireless broadcast; therefore, we did not have to use a base station for USB to I2C conversion, or design a stand-alone board to handle the data

transfer for the WREN. When a user enters a command on the UCA, the CCA receives the user command and wirelessly broadcasts it to all client nodes, which then process the command.

2.7.1.3 Wireless downloader

The wireless downloader application (WDA) is installed on fifteen dedicated downloader nodes using 15 separate channels in the 2.4 GHz ISM band for parallel data download. Each WDA pairs and communicates with the mote client application installed on each *WREN* . Once a WDA and a client are paired, the WDA starts transferring all the data stored on the clients. For reliable in-order delivery of data, the WDA uses a packet-based data transmission protocol (the sliding window protocol), which waits for lost packets before moving onto the next window for data transmission. More detailed information on data collection, data download, and data processing for *WRENSys* can be found in [8]. The code for the download process is also open source and publicly available at [9].

2.7.2 Mobile Applications

2.7.2.1 Mote client

The mote client program (MCP) is installed on the *WREN* nodes for detecting contacts among sensors within a 2-meter range. When a *WREN* receives a radio signal from other sensors, it measures the RSSI value of the signal and stores the value in flash memory. The flash storage of *WREN* can store up to 129,000 contact records, which can contain data from deployments lasting more than 1 week with a duty cycle of 20 seconds, depending on average neighborhood size. Another important functionality of the client sensor is self-test and status indication. Although an LED might not be best suited for all situations, because our deployments are prepared for indoor use most of time, we embedded test code to utilize an RGB LED for determining its status to easily run testing routines before deploying sensors. For example, the red LED means a problem with the sensor and green means it is ready for deployment.

2.7.2.2 Mote sleeper

After we realized that carrying base stations to every deployment site required many personnel and deploying sensors within 30 minutes was unrealistic, we created the mote

sleeper program (MSP). We program the sensors before leaving for a deployment site to make the handout process simpler. MSP is installed on an extra mote, "sleeper node", used prior to and during the beginning of a deployment, which prevents the client sensors from collecting data during transportation by sending small radio packets to the clients. Once the sleeper nodes are removed, the client sensors start collecting contact data. This allows deployment personnel to carry bags of sensors to the site and deploy them quickly without carrying base stations.

2.8 Results and Discussion

We discussed the *WREN* hardware and software design, and covered the implementation for a unique study of human contact networks. The study required large-scale, rapid, and frequent deployments to a dynamic environment. In addition, we designed the system to be used by nontechnical people such that epidemiologists could easily maintain the system and manage deployments. In this section, we describe our results and findings from our deployments, and discuss their implications for future research in this space.

2.8.1 Overall Usage

Over the period of 10 months, from April 2012 to February 2013, we collected about 35 million contacts from 26 different deployments. Most deployments happened at schools, but some were completed during summer camps and various activities.

2.8.2 TelosB Deployments

Our first deployment at a school started in April 2012 with 200 TelosB sensors attached to four base stations, as depicted in Figure 2.3. We deployed 138 sensors at that time and collected about 300,000 contact records. Before the second deployment, to make sure the nodes had enough battery life, we decided to replace all 400 batteries of 200 TelosB nodes. This involved opening up the case for each sensor, removing old batteries, putting the new ones in, and then finally closing the lid with a new tape. The whole process took 2 full working days for 2 people to finish. We used the TelosB in 3 total deployments before we were ready with a better solution, the Irene nodes, that would reduce the battery hassle and decrease node maintenance.

2.8.3 Irene Deployments

After the first 3 deployments, we switched to 200 Irene nodes. The new platform necessitated only a few software changes thanks to the similarity of the TelosB and Irene nodes, and the modularization of the code components in TinyOS. The Irene sensors are different in many aspects such as size, weight, and clip design, as shown in Figure 2.4. The main difference between the TelosB and Irene is the charging capability, which helps to alleviate the maintenance concerns. The Irene uses a Lithium button cell battery with a custom-designed charger that fits into the Irene case. The disadvantage of the Irene nodes is their price, totaling $200 when including the case and additional charging mechanism.

Irenes were used for several deployments and summer camps as the case was significantly more robust than the custom acrylic case of the TelosB. However, the Irene sensors showed several drawbacks throughout the course of the deployments. Several charging ports and the USB connectors on the base stations broke after each deployment. We reinforced the connectors with epoxy glue, but still found that 10 out of 200 sensors occasionally broke. Part of the problem came from the Mini USB connector used on the charging circuit, which was rated to only 500 insertions, and was a surface mounted part. The other problem came from the mechanical stress the cable extorted on the case during plugin/plugout events and transport of the nodes. Clearly, a new node had to be designed addressing these issues in order to scale to the thousands of nodes we planned to deploy during the coming winter period.

2.8.4 WREN Deployments

After several Irene deployments, we were finally ready to use our new *WREN* sensors with a total of 20 base stations, each holding 100 individual sensors for charging and programming. As explained in Section 2.6.1, the *WRENSys* improves significantly on ease of maintenance, weight of base stations and nodes, as well as ease of programming the whole system.

The ease of maintenance had a direct impact on the number of personnel needed to deploy the system. The previous TelosB and Irene deployments needed 4 or more people to deploy just 200 sensors over nearly an hour. This dropped to 1 engineering graduate student and 2 assistants easily being able to deploy 400+ sensors in only 30 minutes.

Figure 2.11 shows deployment statistics with the approximate number of contact data collected during each deployment. So far, we made 13 deployments with the *WREN* platform, including the largest deployment of 1524 nodes in February 2013. Participation ranged from 60-100% through our opt-out program, with high-schools having the lowest participation rates, while most schools ranged around 80%.

With *WRENSys* we showed that large-scale, rapid deployments are possible with up to 1500 nodes, and are easily managed by just one graduate student. We believe that the system could scale to a few thousand nodes more, but after that, we hit the limits of practicality. The bottleneck of the system is the process of plugging in each node for recharging and programming. While it takes only 80 seconds to plug in 50 *WREN*s, compared to 5 minutes for the Irene, it is too long once we scale to very large numbers. Wireless programming has been a large topic of research [16] and can be robust and secure. Thus, the largest issue is recharging the battery. Energy harvesting mechanisms could be employed, but the rapid changing environment does not ensure that enough energy can be harvested. We can imagine that a wireless charging base station based on either inductive charging or far-field RF systems could remove the necessity to have a wired base station connection. The wireless charging base station would be designed like a box where nodes can be dropped in to charge regardless of orientation. This would truly make the system easy to maintain.

2.9 Conclusions

In this paper, we described the design and implementation of *WRENSys*, consisting of the new *WREN* nodes, base stations, and application software. Different from regular wireless sensor network deployments, we target rapid, large-scale, mobile, and dynamic systems where nodes move around and infrastructure is nonexistent. We showed successful deployment results for the last 2 years. The ease of deployment and maintenance of *WRENSys* are key to a fast turn-around time of node collection, data download, and redeployment. We used *WRENSys* to study the social networks from elementary to high schools and deployed at over 26 locations, with over 1500 nodes on one location, collecting a total of 35 million contacts between nodes. Our key contributions are, first of all, to meet system requirements for the large-scale human contact network study and to provide a

rapid, large-scale, low-maintenance sensing system with design approaches that set the ground work for further research. The possible future applications are not limited to social contact network study, but can include other applications where the goal of rapid and large-scale deployment is paramount to the system requirements.

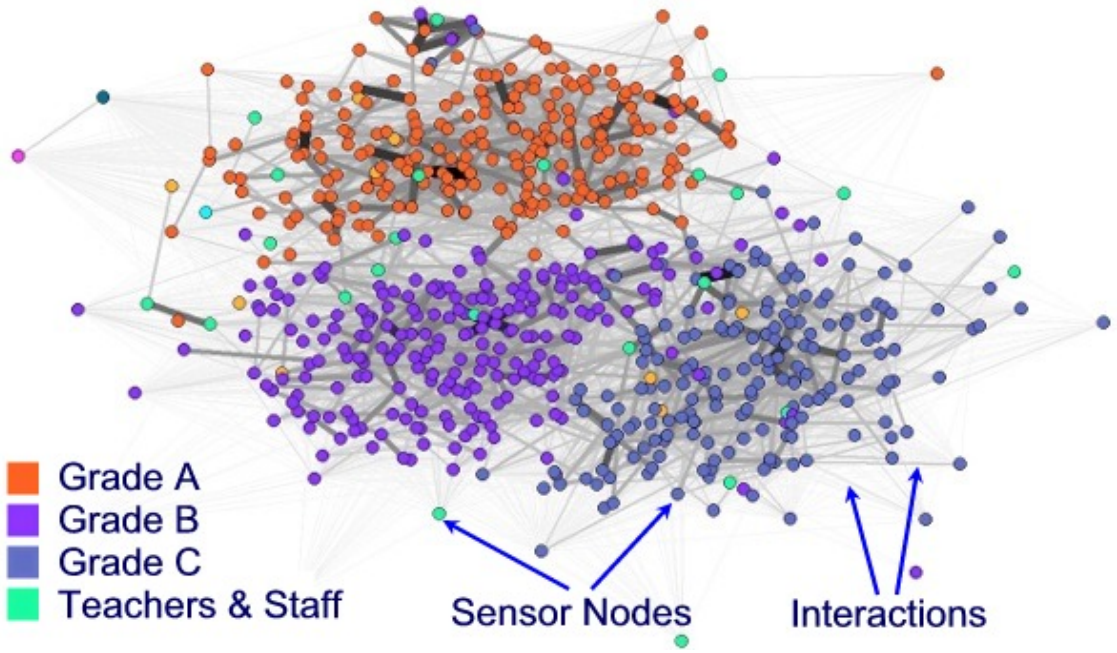


Figure 2.1. Human contact network graph showing data and complex contact patterns between students during a typical school day. The WREN platform collected these data sample. Circles represent sensor nodes and lines represent their interactions. Different colors represent the various grade levels as well as teachers and staff while line thickness represents contact intensity. Epidemiologists access these data to study the spread of airborne diseases.

Figure 2.2. Wireless nodes representing the three hardware iterations. From left to right: TelosB (Memsic), Irene (Moteware), and WREN (our design). They iteratively improve on size, weight, cost, and usability. Enclosures (b). Left: custom laser-cut acrylic case for the TelosB (3.55 cm x 2.54 cm x 7.11 cm). We used tape to hold the case together and to protect the wearer from sharp edges. Middle: standard Irene case. Right: the injection molded WREN case, making it smaller, safer, and more durable (3.3 cm x 1.52 cm x 5.33 cm).

Table 2.1. Summary of the comparison between TelosB, Irene, and WREN platform. We can see that WREN platform stands out in cost, size, and weight.

Node	TelosB	Irene	WREN
Cost ($)	72	200	35
Size (cm)	3.5 x 2.5 x 7.1	5.0 x 1.7 x 6.8	3.3 x 1.5 x 5.3
Weight (g)	83	33	30

Figure 2.3. TelosB nodes with base station and case. The TelosB was worn around the neck using an easy-break safety lanyard, though study subjects often complained about the total weight. A large 50-port USB base station connects the TelosB to a computer for programming and data download. The TelosB were dropped from our deployments due to complications in programming (only around 150 nodes of 200 actually enumerated, even using multiple USB buses), weight of nodes and base stations, and the nonrechargeable batteries.

Figure 2.4. Irene nodes with base station and case. After study subjects' disapproval of lanyards, we designed wrist bands for the Irene nodes. The base station is the same as the TelosB with Micro USB instead of Female USB plugs. The Irene has a rechargeable Li+ battery, removing the need for replacing batteries. This made maintenance easier, but the high price and mechanical reliability issues didn't allow us to scale beyond 200 nodes. Almost every USB connector on the charging circuits had to be epoxied due to breakage after several insertions.

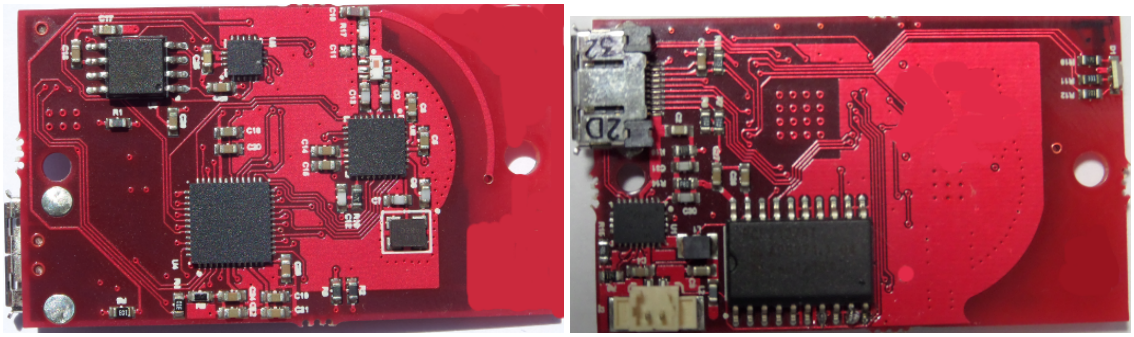


Figure 2.5. Top side of WREN showing radio, MCU, accelerometer, and flash memory. Bottom side showing RTC with integrated 32 kHz TCXO, switch-mode power regulator and battery charger, battery connector, and 10-pin high-reliability Hirose connector.

Table 2.2. Comparison of radio ICs.

	Atmel AT86RF233	TI CC2420
RX	11.8 mA -101 dBm	19.7 mA -94 dBm
TX	13.8 mA 4 dBm	17.4 mA 0 dBm
Max data rate	2 Mbps	250 kbps

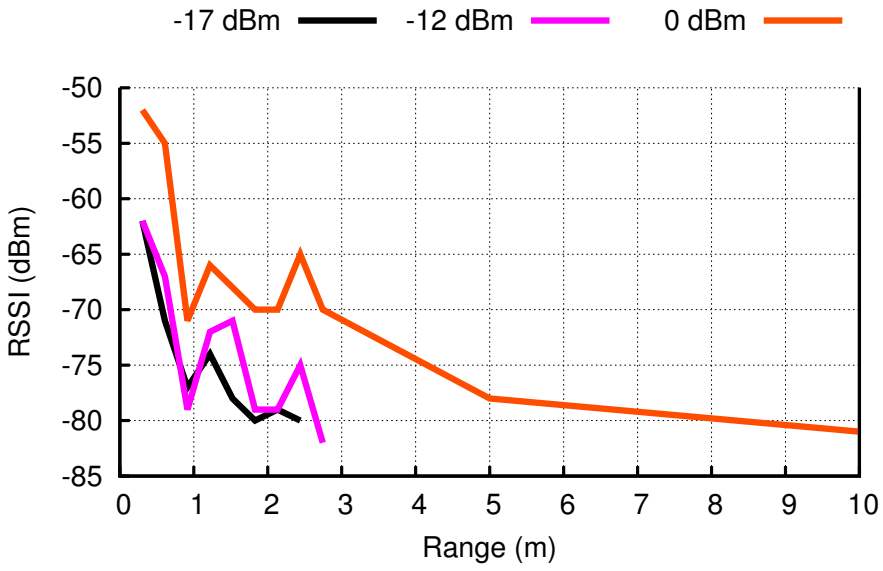


Figure 2.6. Range vs. RSSI. We desensitized the receiver to reduce power draw and range. Without these modifications, the minimum distance was 17 meters. Desensitizing the receiver reduced this range to match the requirements for our deployments.

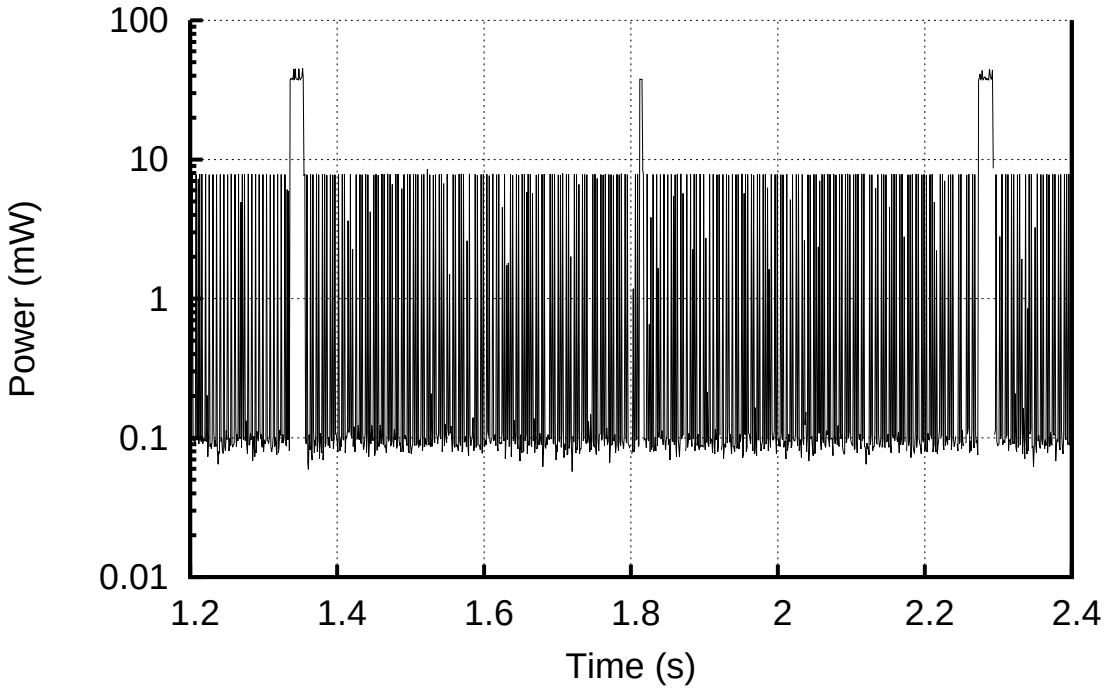


Figure 2.7. WREN power draw for a 0.5 s sleep interval. The average measured power is 5.2 mW, sleep at 60 μW, standard operation at 8 mW, radio actively transmitting 42 mW. The power supply quiescent power is 50 μW. This corresponds to the expected power draws given by the component datasheets.

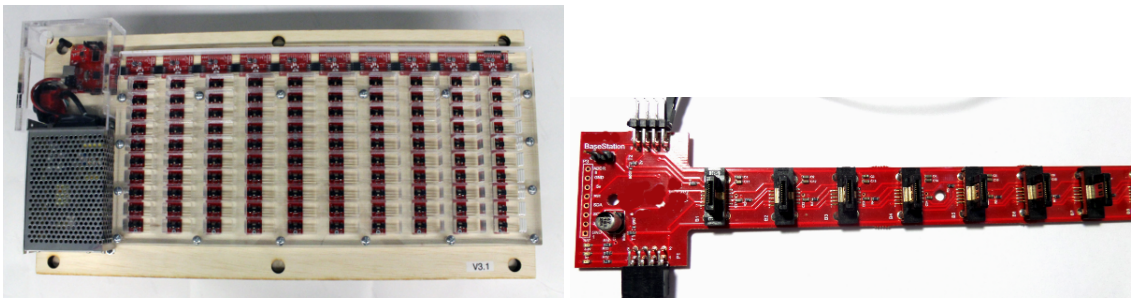


Figure 2.8. Left: Base station structure showing 10 blades, power supply, and status LEDs. Using I2C in extended addressing mode, we could have connected up to 1000 nodes on one station, but power limits for charging, and signal integrity over multiple stations prevented us to go much larger than 100 nodes. Right: A single blade of WREN base station containing the high-reliability 10-pin male Hirose connectors. The modular blades allow us to easily chain blades together for a variable node-sized base station. We were eventually limited by the power supply and trace widths for current, as well as maintaining proper signal integrity for the parallel I2C bus.

Figure 2.9. Base stations with WREN nodes plugged in. We designed the base stations specifically for ease of use, storage, and maintenance in mind. The base stations can be stacked to save space. LEDs indicate the proper function of each blade and their status. The WREN features an RGB LED for debugging and as a status indicator for charging/fully charged while plugged in. These LED make it simple to identify broken nodes, thus reducing the danger of redeploying nonfunctioning systems.

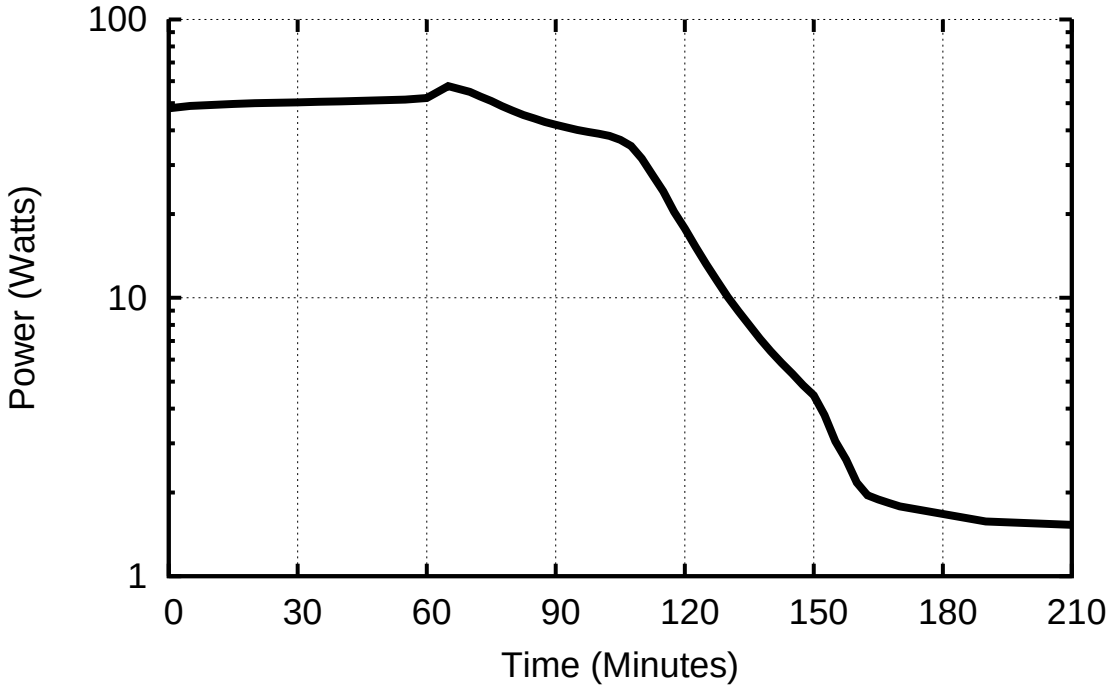


Figure 2.10. Power usage graph of fully populated base station. The base station charges 100 nodes within 2.5 hours using 50 W. Each WREN draws around 100 mA at 5 V during charging. Thus, our power supply and blades had to carry up to 10 A of current on the power traces.

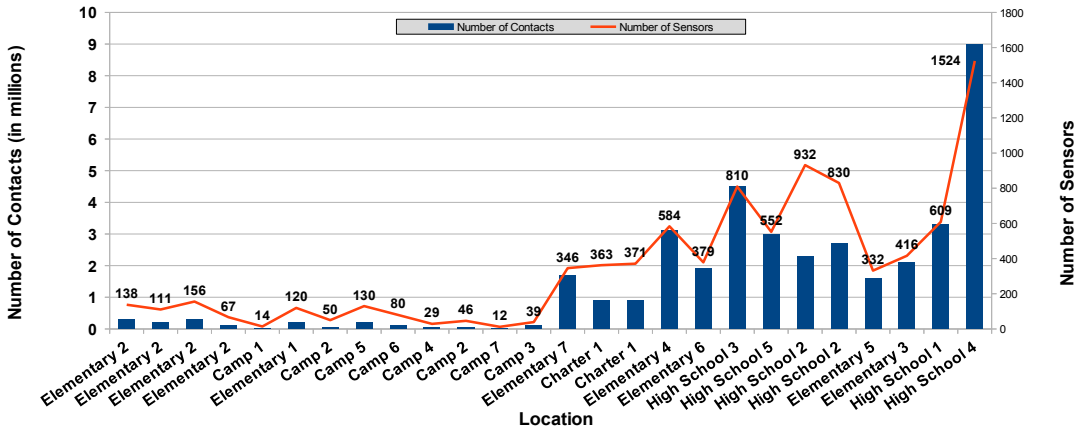


Figure 2.11. Deployment results showing the number of sensors deployed and the number of contacts measured in school and camp deployments from 2012 to 2013. The 13 deployments with less than 200 sensors were performed using TelosB and Irene nodes. The larger deployments of more than 200 nodes was done using the WREN sensors. This was possible due to the reduced cost of the nodes and lower maintenance and programming overhead.

2.10 References

[1] CDC, "Cdc reports about 90 percent of children who died from flu this season not vaccinated," http://www.cdc.gov/flu/spotlights/children-flu-deaths.htm, March 2013.

[2] "How flu spreads," http://www.cdc.gov/flu/about/disease/spread.htm, September 2013.

[3] Simon Thompson, "Localization," *Journal of Comparative and Physiological Psychology*, vol. 69, no. 4, pp. 1–29, 1969.

[4] Ahmad Rahmati et al., "Exploring iphone usage: the influence of socioeconomic differences on smartphone adoption, usage and usability," in *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*. ACM, 2012, pp. 11–20.

[5] Nathan Eagle and Alex Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[6] Maria A Kazandjieva et al., "Experiences in measuring a human contact network for epidemiology research," in *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*. ACM, 2010, p. 7.

[7] Andrzej Forys et al., "Rapid deployable system for human contact network research," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 383–384.

[8] Andrzej Forys, Kyeong Min, Thomas Schmid, Warren Pettey, Damon Toth, and Molly Leecaster, "Wrenmining: Large-scale data collection for human contact network research," in *Proceedings of First International Workshop on Sensing and Big Data Mining*. ACM, 2013, pp. 1–6.

[9] "Utah-wiesel github," https://github.com/Utah-WiESEL/wren_wsn.

[10] Marcel Salathé et al., "A high-resolution human contact network for infectious disease transmission," *Proceedings of the National Academy of Sciences*, vol. 107, no. 51, pp. 22020–22025, 2010.

[11] "OpenBeacon," http://www.openbeacon.org, March 2013.

[12] "SocioPatterns," http://www.sociopatterns.org, March 2013.

[13] Juliette Stehlé et al., "High-resolution measurements of face-to-face contact patterns in a primary school," *PloS one*, vol. 6, no. 8, pp. e23176, 2011.

[14] Yunhao Liu et al., "Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs," *Frontiers of Computer Science in China*, vol. 4, no. 3, pp. 334–338, 2010.

[15] Prabal Dutta et al., "A building block approach to sensornet systems," in *SenSys'08: Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems*, nov 2008.

[16] Luca Mottola and Gian Pietro Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 19, 2011.

PART II

DEVICE-FREE LOCALIZATION

CHAPTER 3

DIAL IT IN: ROTATING RF SENSORS TO ENHANCE RADIO TOMOGRAPHY

3.1 Abstract

A radio tomographic imaging (RTI) system uses the received signal strength (RSS) measured by RF sensors in a static wireless network to localize people in the deployment area, without having them carry or wear an electronic device. This paper addresses the fact that small-scale changes in the position and orientation of the antenna of each RF sensor can dramatically affect imaging and localizaftion performance of an RTI system. However, the best placement for a sensor is unknown at the time of deployment. Improving performance in a deployed RTI system requires the deployer to iteratively "guess-and-retest", *i.e.*, pick a sensor to move and then re-run a calibration experiment to determine if the localization performance had improved or degraded. We present an RTI system of *servo-nodes*, RF sensors equipped with servo motors which autonomously "dial it in", *i.e.*, change position and orientation to optimize the RSS on links of the network. By doing so, the localization accuracy of the RTI system is quickly improved, without requiring any calibration experiment from the deployer. Experiments conducted in three indoor environments demonstrate that the servo-nodes system reduces localization error on average by 32% compared to a standard RTI system composed of static RF sensors. [1]

3.2 Introduction

Radio tomographic imaging (RTI) systems [1], [2] localize and track people in indoor areas using the received signal strength (RSS) measurements made by a mesh network of

[1] In this joint work, I designed and implemented the hardware and firmware to automatically moved sensors in order to optimize the fading condition of the channels. I was also a part of deployment and testing of the full localization system. This section of the chapter appears in 2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), "Dial It In: Rotating RF Sensors to Enhance Radio Tomography", Maurizio Bocca, Anh Luong, Neal Patwari, and Thomas Schmid.

multiple static wireless transceivers. These devices are called "RF sensors" because their RF interface is their mode of sensing. Instead of requiring people to carry or wear an electronic device (*e.g.*, RFID tag, mobile phone, etc.), an RTI system uses the changes in RSS on the network's links to estimate the attenuation field caused by the presence and movements of people found in it. RTI systems can be used to enable context awareness [3]–[6], in ambient-assisted living (AAL) and elder care applications [7]–[9], and in tactical operations or crisis situations [10], [11].

An effect we have observed over many deployments in real-world indoor environments is that the performance of an RTI system can be dramatically altered (either improved or degraded) by small (*i.e.*, sub-wavelength) position changes of the deployed RF sensors. Two RTI deployments in the same area, with RF sensors deployed in ostensibly the same positions, may have significantly different localization and tracking performance. We show an example of how RTI performance is improved by moving one sensor in Section 3.4.1.

One may systematically improve the positions of the RF sensors, and thus the performance of the RTI system, by a long and tedius procedure we call "guess-and-retest". First, RF sensors are deployed so as to form links that cover uniformly the monitored area. An experiment is then conducted with the deployer moving along a known path, by which the tracking error of the RTI system is calculated. Next, the deployer:

1. picks a sensor to be the *sensor-under-test* and changes its position, moving it a few cm in one direction or another.

2. re-performs the known-path experiment and recalculates the tracking error.

3. If the tracking error increases, the deployer moves the sensor-under-test back to its original position.

4. Repeat from Step 1.

While the "guess-and-retest" procedure is possible and will ultimately reduce the localization error, it is extraordinarily time-consuming. Moreover, in highly dynamic indoor environments, such as office spaces or apartments, the same procedure should be periodically performed to re-adjust the positions of the RF sensors. For these reasons, the

"guess-and-retest" procedure is unsuitable for a real-world deployment of a commercial RTI system.

In this paper, we introduce an RTI system composed of a network of autonomously rotating RF sensors, which we call *servo-nodes*, shown in Figure 3.1. Each servo-node is equipped with a servo motor and is capable of performing small-scale, *i.e.*, on the order of a wavelength, adjustments of the position of the RF sensor. Further, we suggest and justify a simple, network-wide quality metric which is based solely on link channel measurements when no person is in the area, and thus does not require the deployer to conduct any known-path experiment. Together, the quality metric and the servo-motor allow each sensor to quickly, *i.e.*, within seconds, "dial it in", *i.e.*, rotate to optimize its own position. We show via three deployments that this procedure, which we refer to as *calibration*, reduces the localization error by 30% to 37%.

The logic behind the proposed calibration procedure and the key to the improvement in localization performance is in the optimization of the links' *fade level*, *i.e.*, the degree to which RSS is changed by constructive or destructive multipath fading. If multipath components arrive at the receiver antenna with nearly the same phase, the link is said to be in *anti-fade*, and its RSS is relatively high. Alternatively, if components have nearly opposite phase, a link is said to be in *deep fade*, and its RSS is relatively low [12]. Since the phase of each component changes at a different rate as the antenna is moved, we observe the effect of small-scale fading [13].

Previous works [7], [12], [14], [15] have demonstrated that the change in RSS induced by a person obstructing the link line, *i.e.*, the straight, imaginary line connecting transmitter and receiver, strongly depends on the fade level of the link. Anti-fade links measure a consistent attenuation only when the person is located in the proximity of the link line. In contrast, deep fade links measure a variation in RSS (either increase or decrease) also when the person is located at unpredictable positions far away from the link line. Thus, anti-fade links provide generally more informative and reliable information about a person's position. We propose that maximizing the sum of RSS measured on all links in empty-room conditions will increase, on average, link fade levels, and thus improve RTI tracking accuracy.

Servo-nodes do not move the RF sensors far enough to alter large scale path loss on

links. Thus, any increase in RSS at the receiver can be attributed only to a change in small-scale fading that makes the multipath phasor sum more constructive. With the link multipath arranged to be more constructive, there is a higher probability that the link will exhibit a more reliable and predictable attenuation behavior when the person is actually obstructing the link line. Therefore, increasing the links RSS will improve the RTI system localization accuracy. Instead of the "guess-and-retest" procedure, the deployer simply turns on and deploys servo-nodes and leaves the room. The RF sensors self-calibrate by rotating to a (local) optimum position. Even better, the servo-nodes can periodically recalibrate their positions to adjust to changing environmental conditions over months and years.

We present results from three deployments, *i.e.*, a typical one bedroom apartment, a highly cluttered university laboratory, and a large office space. Preliminary experiments are conducted in the apartment with a multi-node platform (see Figure 3.2 and Section 3.3.1.1) that simulates the functioning of a servo-node. The servo-nodes (see Figure 3.1 and Section 3.3.1.2) are used in the subsequent (lab and office) deployments. We also describe two different calibration procedures that iteratively adjust position and orientation of the servo-nodes composing the RTI system. Both procedures aim at increasing the overall RSS of the links of the network in static conditions, so as to increase the number of anti-fade links and consequently improve the localization accuracy. The results of the deployments show that a system composed of rotating RF sensors in random positions, *i.e.*, with random orientation, achieves a localization accuracy similar to the one of a *standard* RTI system composed of static nodes all with the same orientation. However, when the servo-nodes are calibrated, the localization error is reduced on average by 30% compared to a standard RTI system with the same number of sensors. Alternatively, the calibrated servo-nodes system achieves the same localization accuracy as a system composed of standard sensors, but with 37% fewer sensors.

3.3 Methods

In this section, we describe the multi-node platform and the rotating servo-node used in the deployments, the procedures applied to calibrate position and orientation of the RF sensors, and the RTI method used to process the RSS measurements collected in the

experiments.

3.3.1 Hardware

3.3.1.1 Multi-node platform

To conduct preliminary experiments, we created a multi-node platform (see Figure 3.2) with eight battery-powered RF sensors attached to a rigid cardboard tile and positioned clockwise every 45 degrees along the perimeter of a circle having a 10 cm radius. Each sensor of the platform has different orientation. The platform was designed to simulate the functioning of the rotating servo-nodes (see Section 3.3.1.2), having an RF sensor in each of the eight positions where the servo motor can position its own sensor.

3.3.1.2 Servo-nodes

The servo-nodes are composed of two parts: the RF sensor, *i.e.*, a TI CC2531 USB dongle [16], and the servo motor, *i.e.*, a GWS digital sail winch servo [17]. The RF sensor operates in the 2.4 GHz ISM band. It has a maximum nominal transmit power of 4.5 dBm and can transmit on one of 16 selectable frequency channels, which are 5 MHz apart, as specified by the IEEE 802.15.4 standard. The servo motor can rotate one full turn (360 degrees) through the standard 1-2 ms pulse width modulation (PWM). A rigid cardboard circle, having a 10 cm radius, is glued to the winch of the servo motor. The RF sensor is in turn attached to the cardboard circle so that its antenna is perpendicular to the surface of the circle. The CC2531 platform controls the position of the servo motor through one of its I/O ports. We programmed the nodes so to be able to rotate them to eight different positions ($p = \{1, ..., 8\}$), 45 degrees apart.

The RF sensors collect RSS measurements on the selected frequency channels by running the *multi-Spin* communication protocol [7]. The packets broadcasted by the servo-nodes and received at the central sink node include the RSS measurements of the links of the network and indicate the current position of the servo-nodes. multi-Spin reserves one slot at the end of each TDMA communication cycle in order for the sink node to communicate a new position to one of the servo-nodes.

3.3.2 RF Sensors Calibration Procedures

We now introduce two different procedures to calibrate the small-scale position and orientation of the RF sensors. The first, which we refer to as *incremental* calibration, was used with the multi-node platform for preliminary experiments carried out in the one bedroom apartment. The second, which we refer to as *network* calibration, was used with the servo-nodes in experiments carried out in the laboratory and office space.

3.3.2.1 Incremental calibration

After positioning the first RF sensor (*i.e.*, #1 in Figure 3.3) in a pre-determined position and orientation, the other RF sensors are calibrated and deployed by applying the following iterative procedure:

1. Pick a spot in the monitored area to temporarily deploy the multi-node platform. The spot is chosen so as to maximize the length of the formed links and cover the whole deployment area uniformly (*e.g.*, one can sequentially use the four cardinal points as a reference);

2. In static conditions, *i.e.*, with no people in the deployment area, measure for a short period of time (*e.g.*, 10 s) the RSS of all the links among the eight RF sensors on the multi-node platform and the RF sensors already calibrated and deployed.

3. For sensor $p \in \{1, ..., 8\}$ on the multi-node platform, calculate $\bar{R}_p$, *i.e.*, the mean of the time-averaged RSS in static conditions of all the links between p and the other sensors $d \in \mathcal{D}$ already calibrated and deployed, as:

$$\bar{R}_p = \frac{1}{|\mathcal{D}|} \frac{1}{|\mathcal{C}|} \sum_{d \in \mathcal{D}} \sum_{c \in \mathcal{C}} (\bar{r}_{(p,d),c} + \bar{r}_{(d,p),c}), \tag{3.1}$$

   where $\mathcal{C}$ is the set of measured frequency channels and $|C|$ its cardinality.

4. Remove the multi-node platform and permanently deploy the RF sensor with the highest value of $\bar{R}_p$ where it was located. Add the newly deployed sensor to the set $\mathcal{D}$. Then, add a new RF sensor to the multi-node platform in place of the deployed one.

5. Repeat step 1) through 4).

3.3.2.2 Network calibration

In the incremental calibration, the RF sensors are calibrated and deployed one at a time. In the network calibration, first, all the RF sensors composing the system are deployed. With the servo motors in their default position ($p = 1$), we measure for a short period of time (*e.g.*, 10 s) the RSS of all the L links of the network in static conditions. At the end of this period, we calculate $\bar{R}$ as the mean of the time-averaged RSS in static conditions of the L links of the network on the frequency channels in $\mathcal{C}$:

$$\bar{R} = \frac{1}{L}\frac{1}{|\mathcal{C}|}\sum_{l=1}^{L}\sum_{c\in\mathcal{C}}\bar{r}_{l,c}. \tag{3.2}$$

Starting from RF sensor #1, we apply to each RF sensor s of the system the following calibration procedure:

1. Collect M RSS measurements ($M = 10$ in our tests) for each link and frequency channel, *i.e.*, collect RSS measurements for the time interval corresponding to M multi-Spin TDMA communication cycles.
2. Calculate and store $\bar{R}_s^p$, *i.e.*, $\bar{R}$ with the servo motor of sensor s in position p.
3. While $p \leq 8$, rotate sensor s to the next position and repeat steps 1) and 2).
4. If $\max(\bar{R}_s^p) > \bar{R}_N$, rotate sensor s to the corresponding position p and set $\bar{R} = \max(\bar{R}_s^p)$. Otherwise, rotate sensor s back to its original position, *i.e.*, the position sensor s had at the end of the last iteration of the calibration procedure.

The calibration procedure is repeated until the rotating RF sensors do not set into different positions compared to the previous iteration, *i.e.*, until $\bar{R}$ does not increase anymore. The iterative approach ensures that all possible permutations of positions of the servo-nodes are evaluated before the servo-nodes set in the final configuration, achieving the maximum value of $\bar{R}$.

For the largest system used in our experiments, composed of 14 servo-nodes, the length of a multi-Spin TDMA communication cycle is approximately 200 milliseconds (the length being proportional to the number of RF sensors). Thus, the time required to collect 10 RSS measurements for each link and frequency channel with the servo-motor in a specific position is approximately 2 seconds. Overall, the calibration of a servo-node takes approximately 15 seconds.

3.3.3 Radio Tomographic Imaging

In this section, we summarize the RTI method, introduced in [15], used to process the RSS measurements collected in the experiments. To the best of our knowledge, the method in [15] is, to date, the RTI method achieving the highest localization accuracy. By processing the data with this method, we aim at demonstrating that a system composed of rotating RF sensors can further enhance the localization accuracy of RTI. While in this work we consider only the method in [15], we expect similar improvements in localization accuracy by using rotating RF sensors also with other RTI methods [14], [18], [19].

An RTI system is composed of N RF sensors deployed at known positions $\{x_n, y_n\}_{n=1,\dots,N}$ and communicating on a set $\mathcal{C}$ of different frequency channels. At each time instant k, the system measures the RSS $r_{l,c}(k)$ of link l on frequency channel $c \in \mathcal{C}$. By combining the RSS measurements collected on all the $L = N \cdot (N-1)$ links of the network on the C selected frequency channels, the system estimates in real-time the change in the propagation field of the monitored area caused by people found in it.

During an initial training phase of the system, performed when the deployment area is not occupied by people, we measure the average RSS of each link on each measured frequency channel. We denote this as $\bar{r}_{l,c}$. After the training phase, we estimate the RSS attenuation of link l on channel c at time instant k as:

$$\Delta r_{l,c}(k) = r_{l,c}(k) - \bar{r}_{l,c}. \tag{3.3}$$

In RTI, the attenuation field to be estimated is discretized into voxels. The attenuation of a link is assumed to be a spatial integral of the RF propagation field of the monitored area. Thus, for each link, the change in RSS is a linear combination of the change in the attenuation of a subset of voxels, *i.e.*, the voxels within an ellipse having the transmitter and receiver of the link at the foci.

While in previous works the width of the ellipse λ was set to a fixed value for all the links of the network, the method in [15] defines a different value λ_l for each link based on its *fade level* [12]. The fade level of link l on channel c is defined as:

$$F_{l,c} = \bar{r}_{l,c} - P(d,c), \tag{3.4}$$

where $P(d,c)$ is the theoretical RSS estimated by using the log-distance path loss model [13], which depends on the distance d between transmitter and receiver and on the center

frequency c. The path loss exponent η of the model is derived after the initial calibration by applying linear least squares fitting to the measured average RSS of all the links of the network. After this, the fade level of the links on each selected frequency is calculated as in (3.4).

As defined in [12], an *anti-fade* link-channel pair (l, c) has positive fade level, while a *deep fade* one has negative fade level. The characteristics of these two types of links have been described in [12] and then modeled in [15]. The work in [12] demonstrated that the sensitivity area of deep fade links is larger than the one of anti-fade links. In addition, when a deep fade link is obstructed, on average the measured RSS increases. Instead, when an anti-fade link is obstructed, on average the measured RSS decreases. The model described in [15] determines for each link-channel pair (l, c) two parameters, λ^+ and λ^-, *i.e.*, the width of the ellipse for a measured increase $(+)$ and decrease $(-)$ in RSS, respectively, based on the fade level. The value of λ^- is considerably smaller for anti-fade links than for deep fade links, as anti-fade links measure a decrease in RSS only when the person is located in the close proximity of the link line. Thus, anti-fade links provide higher quality information for the purpose of device-free localization. By calibrating the small-scale position of the servo-nodes, our system increases the mean RSS of the links of the network, pushing the links towards an anti-fade like behavior, which in turn increases the localization accuracy.

For each link-channel pair, based on the so determined $\lambda^+_{l,c}$ and $\lambda^-_{l,c}$, its fade level, and the magnitude of the change in RSS calculated as in (3.3), the model in [15] derives the probabilities $p^+_{l,c}$ (for an increase in RSS) and $p^-_{l,c}$ (for a decrease in RSS) of the person being within the area defined by the corresponding ellipse: the larger the change in RSS, the higher the probability.

When all the L links of the network are considered, the change in the propagation field of the monitored area can be estimated as:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{n}, \tag{3.5}$$

where $\mathbf{y}$ and $\mathbf{n}$ are the measurements and noise vectors and $\mathbf{x}$ is the image to be estimated. The measurements vector is composed of the probabilities $p^+_{l,c}$ and $p^-_{l,c}$ for each link-channel pair of the network. The elements of the weight matrix $\mathbf{W}$, representing how

the attenuation of the voxels affect the RSS measurements of the links, are calculated as:

$$w_{l,c,j}^{\delta} = \begin{cases} \frac{1}{A_{l,c}^{\delta}} & \text{if } d_{l,j}^{tx} + d_{l,j}^{rx} < d_l + \lambda_{l,c}^{\delta} \\ 0 & \text{otherwise} \end{cases}, \tag{3.6}$$

where $\delta = \{+, -\}$ represents the sign of the change in RSS, $A_{l,c}^{\delta}$ is the area of the ellipse, $d_{l,j}^{tx}$ and $d_{l,j}^{rx}$ are the distances from the center of voxel j to the transmitter and receiver, respectively, and d_l is the length of the link.

Since the number of links L is considerably smaller than the number of voxels of the image, the estimation problem is an ill-posed one, and regularization has to be applied. We use a regularized least-squares approach [20]:

$$\hat{\mathbf{x}} = \Pi \mathbf{y}. \tag{3.7}$$

The projection matrix Π is calculated as:

$$\Pi = (\mathbf{W}^T \mathbf{W} + \mathbf{C}_x^{-1} \sigma_N^2)^{-1} \mathbf{W}^T, \tag{3.8}$$

where σ_N^2 is the regularization parameter. The *a priori* covariance matrix $\mathbf{C}_x$ is calculated by using an exponential spatial decay:

$$[\mathbf{C}_x]_{j,i} = \sigma_x^2 e^{-d_{j,i}/\delta_c}, \tag{3.9}$$

where σ_x^2 is the variance of voxel measurements, $d_{j,i}$ is the distance between the center of voxel j and the center of voxel i, and δ_c is the voxels' correlation distance. The position of a person found in the monitored area is estimated by finding the coordinates of the voxel of the RTI image that has the maximum value.

3.4 Experimental Results

We now present results from three different deployments of our system. In each deployment, we compare the performance of the system composed of servo-nodes to a system composed of *standard* RF sensors, *i.e.*, static sensors all having the same orientation. To make a fair comparison of the performance of the two systems, two standard RF sensors are positioned in the proximity of each servo-node, one on each side, at distance $d \leq 20$ cm from the winch of the servo motor. By so doing, we are ensuring that the links among the nodes of both systems cover the deployment area approximately in the same way, and that the differences in localization accuracy do not originate from the nodes being positioned at different locations.

3.4.1 Effect of Sensors Position on RTI

First, we show how small-scale changes in the position of RF sensors affect the imaging and localization performance of an RTI system. We perform two different simulations by using the same RSS measurements collected with the standard nodes during a test in a 54 m^2 highly cluttered laboratory at the University of Utah (see Figure 3.4(a)). For both simulations (A and B), we select 14 of the 28 deployed standard RF sensors (*i.e.*, one for each of the 14 deployed servo-nodes). 13 of the selected 14 standard sensors are the same in both simulations. However, in simulation A, we select sensor #9, while in simulation B we select sensor #10. These two sensors are 20 cm apart. Figure 3.5 shows the RTI images formed in simulation A and B when the person is located at coordinates $(5.13, 4.57)$ m. The localization error of simulation A is 1.01 m, while the error of simulation B is 0.52 m.

3.4.2 Preliminary Experiments

Preliminary experiments were conducted in a 56 m^2 one bedroom apartment (with floor map shown in Figure 3.3). We calibrated and deployed 13 RF sensors by using the multi-node platform described in Section 3.3.1.1 and the incremental procedure described in Section 3.3.2.1. We also deployed 26 standard sensors, *i.e.*, sensors having the same orientation. For the standard sensors, we chose positions in the proximities of the spots selected for the sensors calibrated with the multi-node platform. All the RF sensors were deployed at approximately 1.2 m from the floor. Both systems, *i.e.*, calibrated and standard, were communicating on the same frequency channels ($\mathcal{C} = \{11, 16, 21, 26\}$). The apartment was located in an area of the building with multiple coexisting Wi-Fi networks (belonging to residents of the neighboring apartments).

To evaluate the localization accuracy, we marked 45 points on the floor of the apartment. These points represented the true position of the person to be localized during a test. We asked the person to stand still at each of these locations for 8 s. In this work, in order to provide a more reliable comparison of the performance of the rotating and standard RTI systems, we consider only the accuracy in localizing a stationary person. However, the methods and systems presented in this work can also be used to localize and track a moving person.

Figure 3.6 shows the box plot of the results of five different tests conducted in the

one bedroom apartment. In each test, the person stood in the 45 evaluation points with different orientations of her body. With the 13 sensors calibrated with the multi-node platform, the median root mean squared error (RMSE) is 0.39 m. For each test, we create 30 different subsets of 13 standard sensors by selecting, for each calibrated sensor, one of the two neighboring standard sensors. By so doing, we ensure that the links connecting the selected sensors cover the deployment area uniformly. Simulations are performed by using the data collected during the tests. The median RMSE of the resulting 150 simulations is 0.62 m. Thus, the system composed of calibrated RF sensors reduces the localization error by 37% compared to a standard system composed of sensors all having the same orientation.

We also perform simulations by increasing the number of sensors composing the standard RTI system. For each test, we create 30 different subsets of standard sensors. In this case, sensors are chosen randomly. The results of the simulations show that the median RMSE decreases with a larger number of sensors. With 19 standard sensors, the median RMSE is 0.39 m, *i.e.*, the same as the median RMSE obtained with 13 calibrated sensors. Thus, by calibrating and deploying the sensors using the multi-node platform, we are able to achieve the same localization accuracy by using 32% fewer sensors.

3.4.3 Servo-nodes Deployments

In this section, we present the results of two deployments in which we used the servo-nodes described in Section 3.3.1.2 and the network calibration procedure described in Section 3.3.2.2.

First, experiments were carried out in a 54 m^2 highly cluttered laboratory at the University of Utah (see Figure 3.4(a)). We deployed 14 servo-nodes. We also deployed two standard RF sensors in the proximity of each servo-node, one on each side, at a 20 cm distance from the winch of the servo motor. In this deployment, the set of used frequency channels $\mathcal{C} = \{15, 20, 25, 26\}$, in order to minimize the interference with multiple coexisting WiFi networks.

To evaluate the localization accuracy, we marked 32 points on the floor. The points were chosen so as to cover all the areas of the laboratory. First, we carried out 10 tests by having the 14 servo-nodes positioned in 10 different permutations of random positions.

The mean RMSE of these tests was 0.59 m. Then, for each test, we created 10 different subsets of 14 standard sensors by selecting, for each servo-node, one of the two neighboring standard sensors. The mean RMSE of the resulting 100 simulations was 0.61 m. Simulations were performed by using the same RSS measurements collected during the tests. These results demonstrate that the RTI system composed of servo-nodes, when these are not calibrated, achieves on average a localization accuracy very similar to the one achieved by a standard RTI system.

Subsequently, we applied the network calibration procedure to the servo-nodes. Figure 3.7 shows the mean RMSE of three different tests performed after each iteration of the calibration procedure. After the first iteration, the mean RMSE is 0.51 m, *i.e.*, a 16% reduction of the localization error achieved with the standard sensors. After the second iteration, the mean RMSE is 0.46 m, *i.e.*, a 25% reduction. After the third and final iteration, the mean RMSE is 0.43 m, *i.e.*, a 30% reduction.

Additional experiments were carried out in a 100 m^2 office space at the University of Utah (see Figure 3.4(b)). We deployed 12 servo-nodes (3 on each side of the space) and 24 standard sensors (two in the proximity of each servo-node, one per side). The nodes density of this deployment (0.12 nodes/m^2) was considerably lower than in other previous works that use RTI methods and low-power sensors operating in the 2.4 GHz ISM band: for example, in [3], [5], [12], [14], [15], the nodes density assumed values at least four times higher. In this deployment, the nodes communicated on frequency channels $\mathcal{C} = \{15, 20, 25, 26\}$.

To evaluate the localization accuracy of both RTI systems, we marked 23 points on the floor of the office space. First, we carried out 8 tests by having the 12 servo-nodes positioned in 8 different permutations of random positions. The mean RMSE of these tests was 0.72 m. For each test, we created 10 different subsets of 12 standard sensors by selecting, for each servo-node, one of the two neighboring standard sensors. The mean RMSE of the resulting 80 simulations was 0.74 m. The simulations were performed by using the RSS measurements collected during the tests. Also in this deployment, the system composed of servo-nodes in random positions and the system composed of standard sensors have very similar localization accuracy.

Figure 3.7 shows the mean RMSE of three different tests performed after each iteration

of the network calibration procedure. The RMSE is 0.62 m after the first iteration, *i.e.*, a 16% reduction of the localization error achieved with the standard sensors, 0.59 m after the second iteration, *i.e.*, a 20% reduction, 0.53 m after the third iteration, *i.e.*, a 28% reduction, and 0.52 m after the fourth and final iteration, *i.e.*, a 30% reduction.

Figure 3.8 shows the results of simulations performed by increasing the number of sensors composing the standard RTI system. We create 150 different subsets of standard sensors for each number of standard nodes. Sensors are chosen randomly. The median RMSE decreases with a higher number of sensors. With 21 standard sensors, the mean RMSE is 0.52 m, *i.e.*, the same mean RMSE measured with 12 calibrated servo-nodes. Thus, the RTI system composed of servo-nodes achieves the same accuracy by using 43% fewer sensors. The results of the three deployments are summarized in Table 3.1.

3.4.4 Position Distribution

Does the calibrated position, *i.e.*, the position at which the servo-nodes are set to after the calibration procedure, have a bias to one direction or another? If so, we might suspect that a certain orientation is better because of the antenna polarization, or that a certain orientation results in a beneficial antenna gain pattern. Alternatively, if the highest position (#5 for our servo-nodes) ends up being chosen more often, we might suspect that the increase in antenna height is really to credit for the improvements in RTI performance.

To address these questions, we plot the histogram of the calibrated positions across all three deployments. In Figure 3.9, we consider the calibrated positions of the multi-node platform used in the one bedroom apartment, and the final positions of the servo-nodes in the laboratory (*i.e.*, after the third iteration) and office space (*i.e.*, after the fourth iteration). We consider a total of 38 calibrated positions: 13, 14, and 12 nodes in the apartment, laboratory, and office space, respectively. The first node in the apartment is fixed.

The histogram in Figure 3.9 does not show any particular bias in any direction. The maximum in the histogram is 8, at positions #4 and #7. If it is true that the eight positions are equally likely (our null hypothesis), using the multinomial distribution [21], we find that 86.9% of the time all of the positions will have eight or less occurrences. Thus, having 8 occurrences of a particular position in our experiments is not evidence to reject the equally likely positions hypothesis, and thus, we see no position bias.

3.5 Related Work

Over the last few years, RTI has quickly become one of the most popular techniques of device-free localization. However, other RSS-based techniques have also proven to be feasible and accurate. The work in [22] introduced device-free passive (DfP) localization, which leverages typical wireless data network deployments and off-the-shelf wireless cards. This technique has been used to localize and track multiple people in cluttered [23] and in large [24] indoor environments.

Fingerprinting methods have been used in [4], [25] to estimate in which *cells* people are located. The work in [25] uses probabilistic methods based on discriminant analysis. However, these methods require a long calibration period, *i.e.*, 15 to 30 minutes. Other works have specifically addressed this issue, aiming at creating a localization system that would not require an extensive training phase to be carried out in static conditions. The work in [26] uses background subtraction methods typical of machine vision to estimate baseline RSS values from measurements collected while the system is already in use and people may be located in the monitored area. In the context of RTI, the work in [3] applied a low-pass filter to the RSS measurements of the links of the network in order to adapt the baseline RSS to the changes in the environment and make the system able to provide accurate position estimates in the long-run in a domestic environment. Our work demonstrates the effect of small-scale position changes of RF sensors on the performance of an RTI system. The rotating servo-nodes and the fade level optimization procedures we present allow setting RF sensors to positions that improve the quality of the links' RSS measurements. Moreover, through the servo-motors these positions can be quickly adjusted whenever the conditions of the monitored environment change. Servo-nodes and fade level optimization may be useful not only in RTI but also in fingerprinting methods.

Other systems use the time-of-flight (ToF) of radio signals to perform the localization task. The work in [27] exploits the fact that, similar to the RSS, also the ToF of radio signals is affected by a person obstructing the link line. Thus, ToF measurements are used to form RTI images and localize the person in the monitored area. The system in [28] consists of a single device with one antenna for transmission and three for reception. The device transmits a radio signal and then measures the ToF of the signals reflected by the person's

body. A geometric reference model is then used to map the ToF measurements of the receiving antennas to the position of the person.

3.6 Conclusion

The small-scale position of RF sensors significantly affects the performance of an RTI system. A "good" position for each sensor can not be known *a priori*. In this paper, we present an RTI system composed of servo-controlled RF sensors which rotate in a 10 cm radius. Each sensor in the network rotates, iteratively, and adjusts its small-scale position to increase the average RSS measured on links network-wide. By so doing, the average link fade level is increased, improving the RTI estimates. This automated position calibration, which we refer to as "dialing it in", does not require any known-path experimentation from the deployer and occurs within seconds. We demonstrate the system in three experimental deployments and show that it can reduce localization error by over 30% compared to a standard system based on naïve sensor placement. In future work, we will consider using a platform with multiple, electrically switchable antennas, such as the prototype in [29]. The calibration procedure presented in this work can be applied to select for each RF sensor the antenna improving the quality of the links' RSS measurements.

Figure 3.1. Rotating servo-node platform. The automated prototype used in this work is composed of a TI CC2531 RF sensor, operating in the 2.4 GHz ISM band, and a servo motor that can rotate one full turn (360 degrees). The RF sensor is glued on a rigid cardboard disc having a 10 cm radius. The disc is in turn glued on the winch of the servo motor. The sensor controls the position of the servo motor through one of its I/O ports.

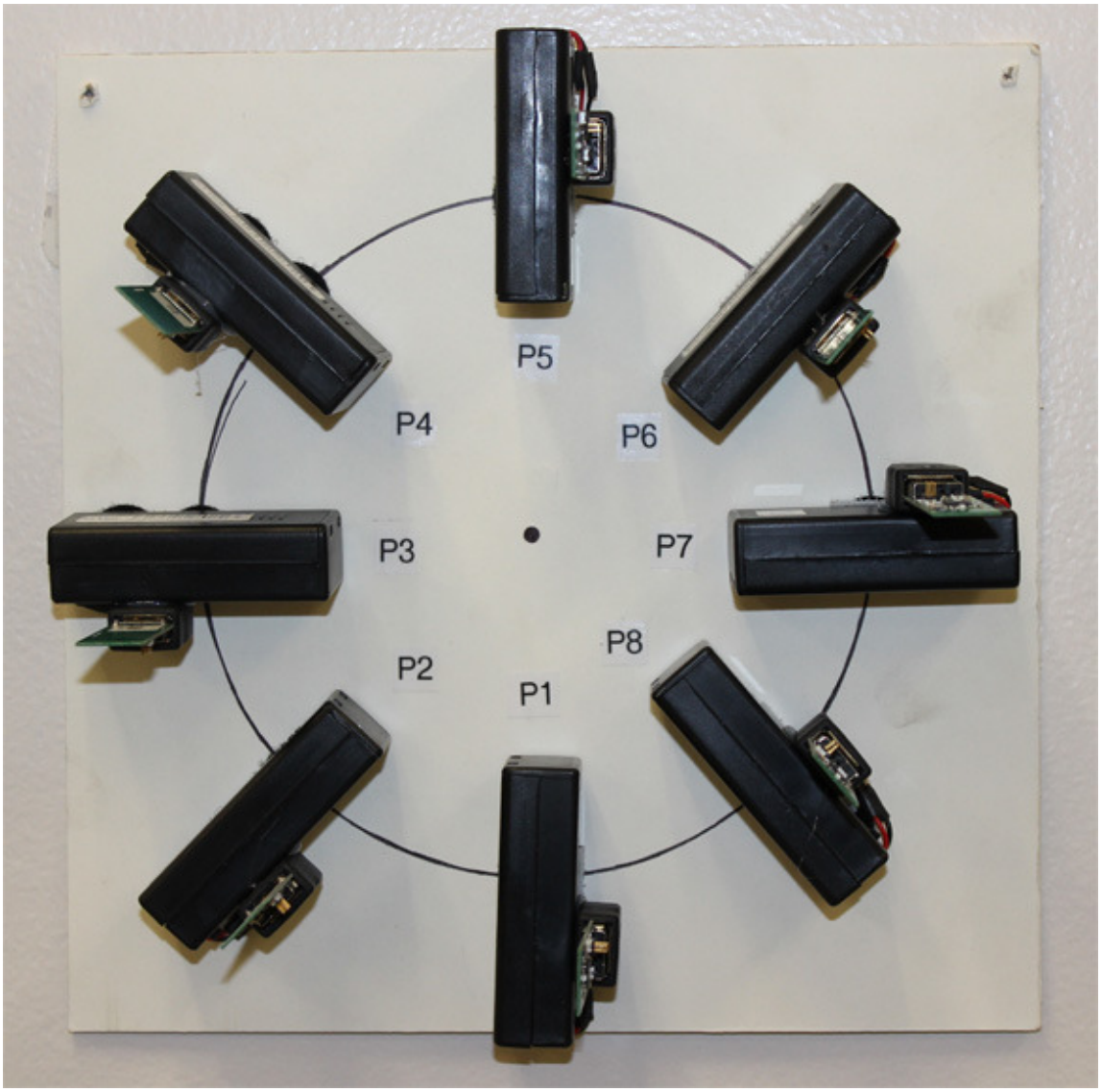


Figure 3.2. Multi-node platform. A prototype is built with eight battery-powered RF sensors positioned clockwise every 45 degrees along the perimeter of a circle having a 10 cm radius. Each sensor has different orientation.

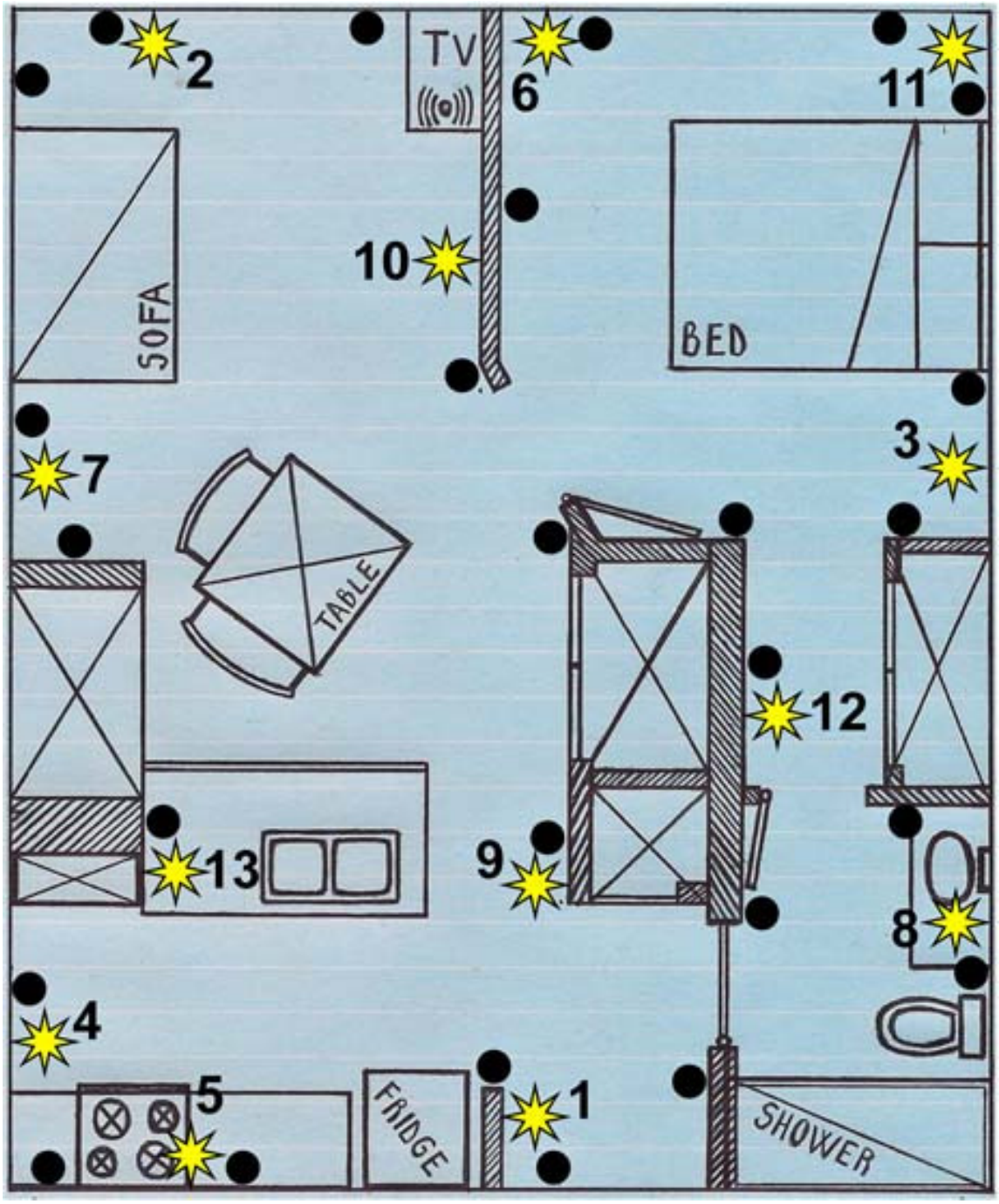
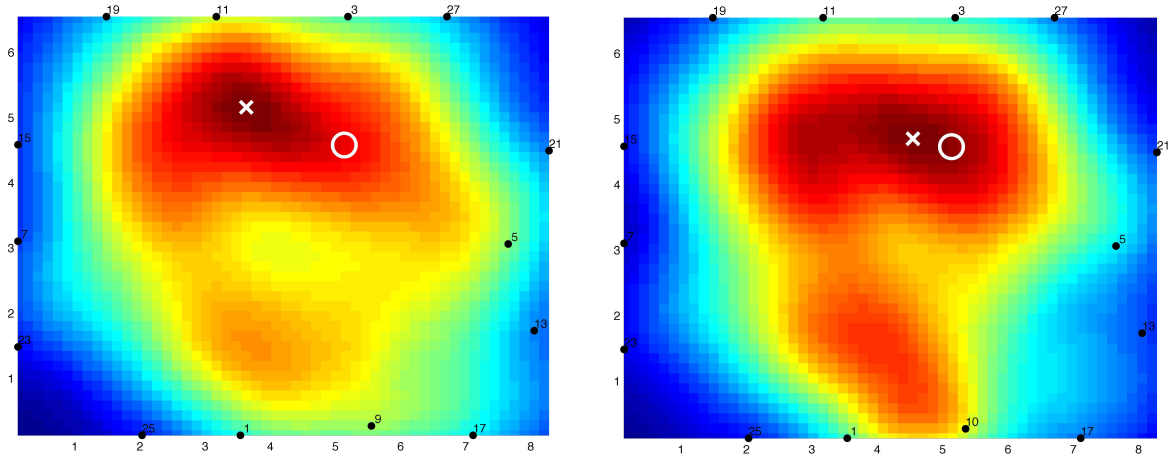


Figure 3.3. Floor map of the one bedroom apartment where preliminary experiments were carried out. The yellow stars represent the RF sensors calibrated with the multi--node platform. The black dots represent the standard RF sensors (all having the same orientation).

(a) Laboratory at the University of Utah (b) Office space at the University of Utah

Figure 3.4. Servo-nodes deployments: in (a), the 54 m^2 laboratory. In (b), the 100 m^2 office space.



(a) Simulation *A* (b) Simulation *B*

Figure 3.5. The effect of sensors' position. RTI images formed by an RTI system composed of 14 standard sensors deployed in a 54 m^2 highly cluttered laboratory at the University of Utah. In simulation *A*, sensor #9 is selected, and the localization error is 1.01 m. In simulation *B*, sensor #10 is selected, and the localization error is 0.52 m. The two sensors are 20 cm apart. In the images, the white circle represents the true position of the person, the white cross the estimated position.

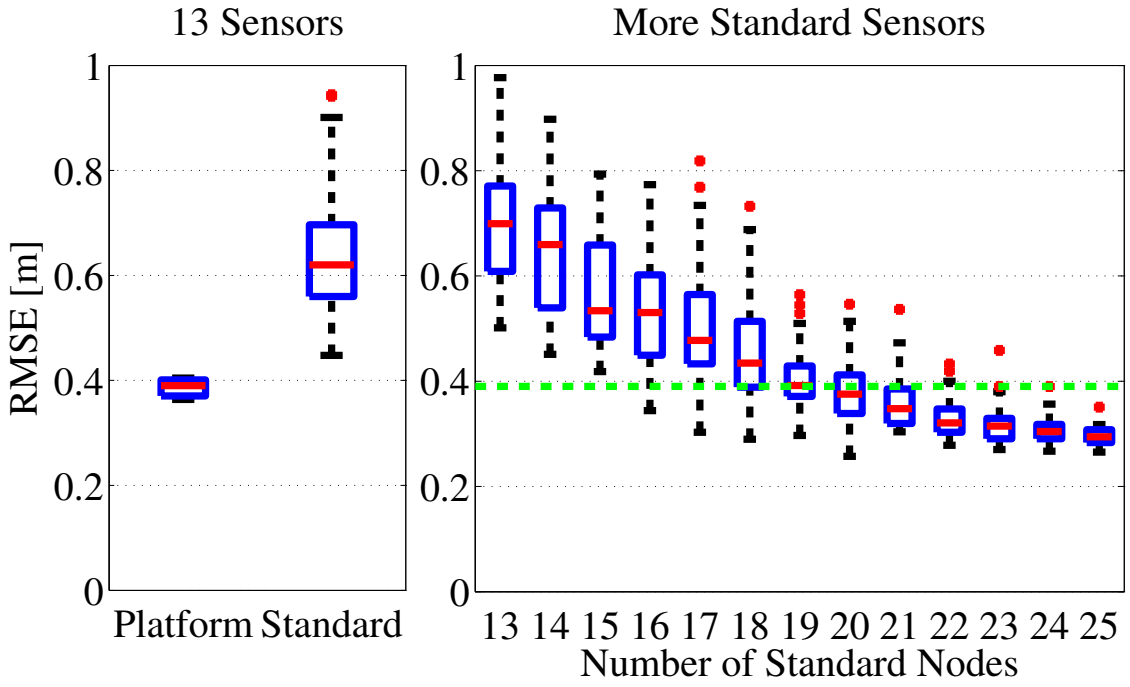


Figure 3.6. Results of the experiments in a 56 m^2 one bedroom apartment. On the left, the RMSE of the system composed of 13 sensors calibrated with the multi-node platform is compared to the RMSE measured with different subsets of 13 standard sensors. On the right, the RMSE measured with a varying number of standard nodes. The horizontal line represents the RMSE measured with 13 sensors calibrated with the multi-node platform.

Table 3.1. Summary of the results

			RMSE [m]				
Deployment	Area [m^2]	# of Nodes	Standard Sensors	Servo-nodes Random Pos.	Servo-nodes Default Pos.	Calibrated Positions	Improvement
Apartment	56	13	0.62			0.39	37%
Laboratory	54	14	0.61	0.59	0.60	0.43	30%
Office space	100	12	0.74	0.72	0.70	0.52	30%
						Average:	**32%**

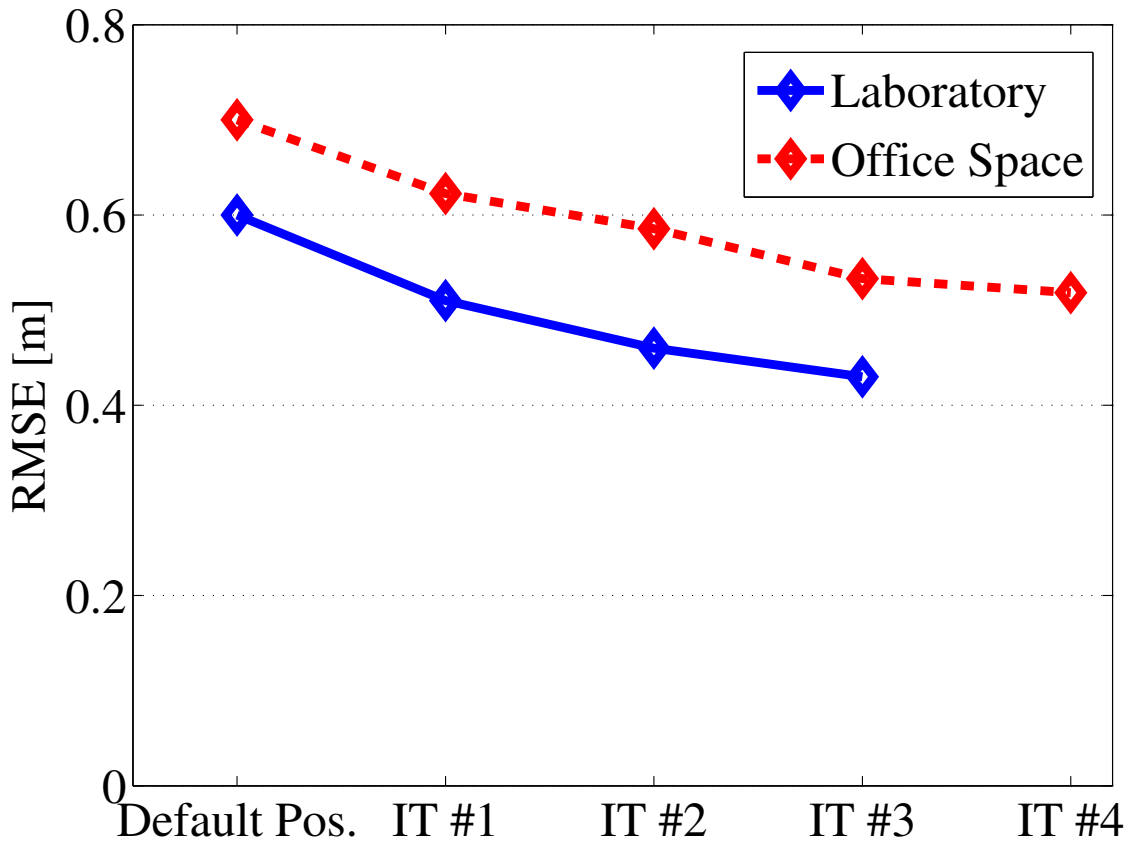


Figure 3.7. Results of the two servo-nodes deployments. In both deployments, the RMSE decreases after each iteration of the network calibration procedure. In the laboratory (blue line), the RMSE with the servo-nodes in default position ($p = 1$) is 0.60 m. After the third and final iteration, the RMSE is 0.43 m. In the office space (red dashed line), the RMSE with the servo-nodes in default position is 0.70 m. After the fourth and final iteration, the RMSE is 0.52 m. By calibrating the RF sensors, we achieve a 30% reduction of the localization error in both deployments.

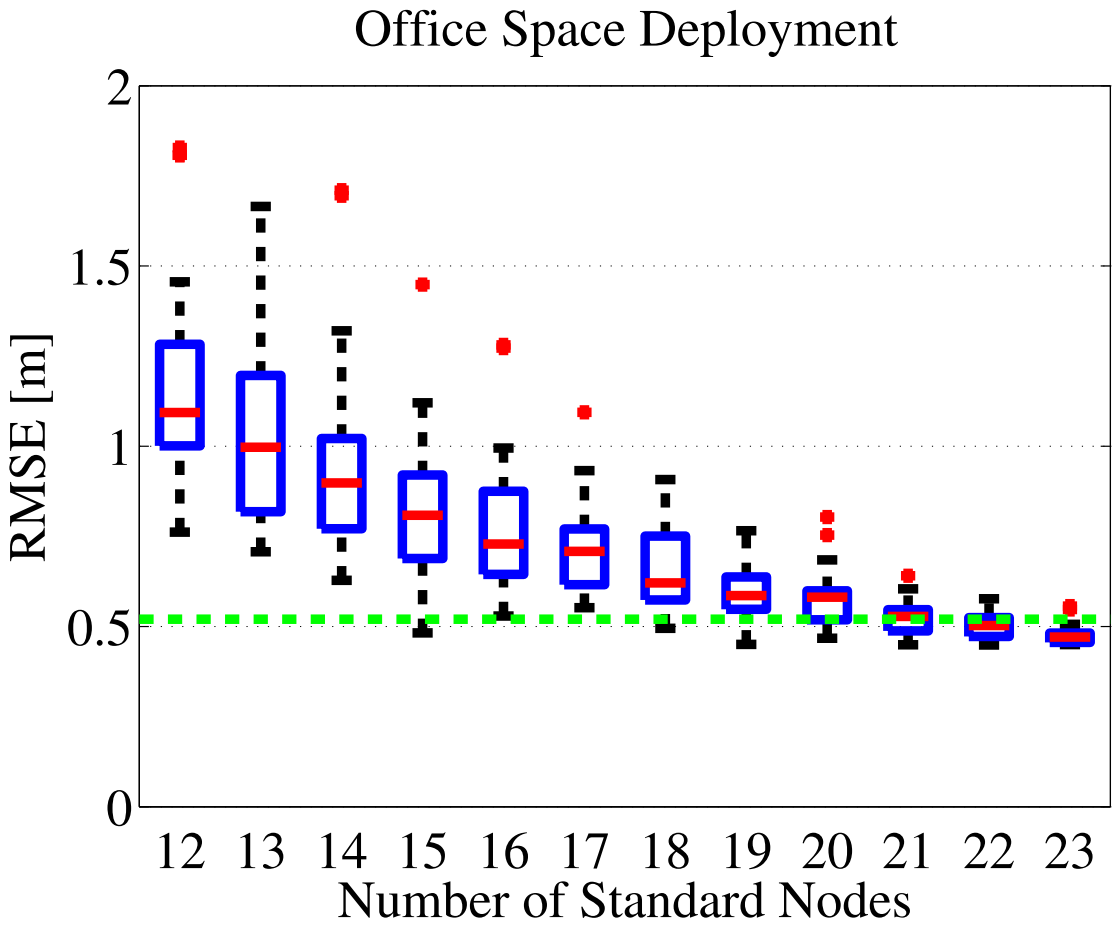


Figure 3.8. RMSE measured with a varying number of standard sensors in a 100 m^2 office space at the University of Utah. The horizontal dashed line represents the RMSE measured with 12 servo-nodes after the fourth and final iteration of the calibration procedure.

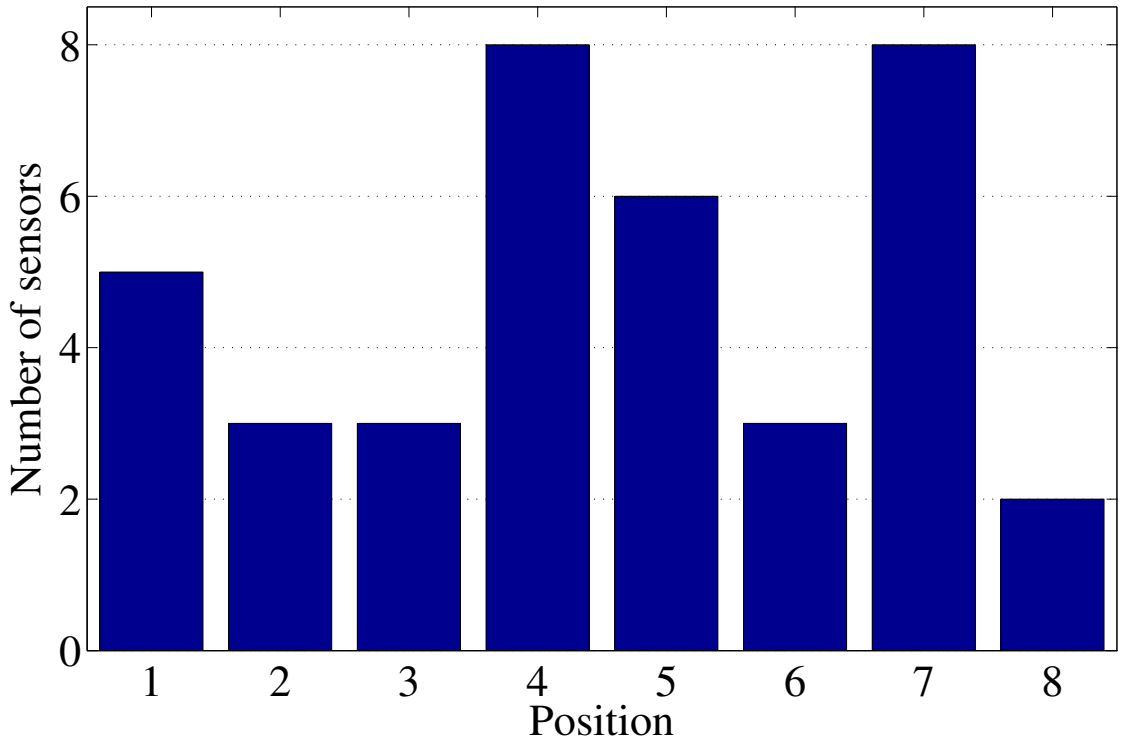


Figure 3.9. Distribution of the calibrated positions of the RF sensors in the three deployments.

3.7 References

[1] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 621 –632, May 2010.

[2] Neal Patwari and Joey Wilson, "RF sensor networks for device-free localization and tracking," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1961–1973, Nov. 2010.

[3] O. Kaltiokallio, M. Bocca, and N. Patwari, "Follow @grandma: Long-term device-free localization for residential monitoring," in *7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp 2012)*, October 2012, pp. 991–998.

[4] Chenren Xu, Bernhard Firner, Robert S. Moore, Yanyong Zhang, Wade Trappe, Richard Howard, Feixiong Zhang, and Ning An, "SCPL: indoor device-free multi-subject counting and localization using radio signal strength," in *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN 2013)*, 2013, pp. 79–90.

[5] Maurizio Bocca, Ossi Kaltiokallio, Neal Patwari, and Suresh Venkatasubramanian, "Multiple target tracking with RF sensor networks," *IEEE Transactions on Mobile Computing*, appeared online July 25, 2013.

[6] Santosh Nannuru, Yunpeng Li, Yan Zeng, Mark Coates, and Bo Yang, "Radio-frequency tomography for passive indoor multitarget tracking," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2322–2333, 2013.

[7] Maurizio Bocca, Ossi Kaltiokallio, and Neal Patwari, "Radio tomographic imaging for ambient assisted living," in *Evaluating AAL Systems Through Competitive Benchmarking*, Stefano Chessa and Stefan Knauth, Eds., vol. 362 of *Communications in Computer and Information Science*, pp. 108–130. Springer Berlin Heidelberg, 2013.

[8] Brad Mager, Neal Patwari, and Maurizio Bocca, "Fall detection using rf sensor networks," in *IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2013)*, September 2013, pp. 3472–3476.

[9] N. Patwari, L. Brewer, Q. Tate, O. Kaltiokallio, and M. Bocca, "Breathfinding: A wireless network that monitors and locates breathing in a home," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 1, pp. 30–42, Feb 2014.

[10] Jussi Timonen and Jouko Vankka, "Enhancing situational awareness by means of visualization and information integration of sensor networks," *Proc. SPIE*, vol. 8756, pp. 87560S–87560S–14, 2013.

[11] Merrick McCracken, Maurizio Bocca, and Neal Patwari, "Joint ultra-wideband and signal strength-based through-building tracking for tactical operations," in *IEEE Int. Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2013)*, 2013, pp. 79–90.

[12] Joey Wilson and Neal Patwari, "A fade-level skew-Laplace signal strength model for device-free localization with wireless networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 947–958, June 2012.

[13] Theodore S. Rappaport, *Wireless Communications: Principles and Practice*, IEEE Press, Piscataway, NJ, USA, 1st edition, 1996.

[14] O. Kaltiokallio, M. Bocca, and N. Patwari, "Enhancing the accuracy of radio tomographic imaging using channel diversity," in *IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, October 2012, pp. 254–262.

[15] Ossi Kaltiokallio, Maurizio Bocca, and Neal Patwari, "A fade level-based spatial model for radio tomographic imaging," *IEEE Transactions on Mobile Computing*, appeared online December 12, 2013.

[16] Texas Instruments, "A USB-enabled system-on-chip solution for 2.4 GHz IEEE 802.15.4 and ZigBee applications," .

[17] GWS, "GWS S125 1T 2BB Sail Winch Servo," .

[18] J. Wilson and N. Patwari, "See-through walls: Motion tracking using variance-based radio tomography networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 612 –621, May 2011.

[19] Yang Zhao, Neal Patwari, Jeff M. Phillips, and Suresh Venkatasubramanian, "Radio tomographic imaging and tracking of stationary and moving people via kernel distance," in *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN 2013)*, 2013, pp. 229–240.

[20] Neal Patwari and Piyush Agrawal, "Effects of correlated shadowing: Connectivity, localization, and RF tomography," in *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN'08)*, Apr. 2008, pp. 82–93.

[21] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, Wiley, 3rd edition, 2000.

[22] Moustafa Youssef, Matthew Mah, and Ashok Agrawala, "Challenges: device-free passive localization for wireless environments," in *ACM Int. Conf. on Mobile Computing and Networking (MobiCom '07)*, 2007, pp. 222–229.

[23] Ibrahim Sabek and Moustafa Youssef, "Multi-entity device-free wlan localization," in *IEEE Global Communications Conference 2012 (GLOBECOM 2012)*, December 2012, pp. 2018–2023.

[24] Moustafa Seifeldin, Ahmed Saeed, Ahmed E. Kosba, Amr El-Keyi, and Moustafa Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1321–1334, 2013.

[25] Chenren Xu, Bernhard Firner, Yanyong Zhang, Richard Howard, Jun Li, and Xiaodong Lin, "Improving rf-based device-free passive localization in cluttered indoor environments through probabilistic classification methods," in *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN 2012)*, 2012, pp. 209–220.

[26] Andrea Edelstein and Michael Rabbat, "Background subtraction for online calibration of baseline RSS in RF sensing networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2386–2398, 2013.

[27] Jie Wang, Qinghua Gao, Hongyu Wang, Yan Yu, and Minglu Jin, "Time-of-flight-based radio tomography for device free localization," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2355–2365, 2013.

[28] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C. Miller, "3d tracking via body radio reflections," in *Usenix NSDI'14 - to appear*, (available online: http://witrack.csail.mit.edu/witrack-paper.pdf).

[29] Erik Österom, Luca Mottola, Martin Nilsson, and Thiemo Voigt, ," in *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN 2010)*. pp. 438–439, ACM.

CHAPTER 4

HIGHLY RELIABLE SIGNAL STRENGTH-BASED BOUNDARY CROSSING LOCALIZATION IN OUTDOOR TIME-VARYING ENVIRONMENTS

4.1 Abstract

Detecting and locating outdoor boundary crossing events is valuable information in curbing drug trafficking, reducing poaching, and protecting high-asset equipment and goods. However, boundary sensing is notoriously challenging, prone to false alarms and missed detections, with serious consequences. Weather events, like rain and wind, make it even more challenging to maintain a low level of missed detections and false alarms. In this paper, we propose and test an automated system of wireless sensors which uses received signal strength (RSS) measurements to localize where a boundary crossing occurs. In addition, we develop new RSS-based statistical models and methods that can quickly be initialized and updated on-line by using link RSS statistics to adapt to time-varying RSS changes due to weather events. These models are implemented in two new classifiers that localize boundary crossings with few missed detections and false alarms. We validate our proposed methods by implementing one of the classifiers in a three month long deployment of a solar-powered, real-time system that captures images of the boundary for ground truth validation. Furthermore, over 75 hours of RSS measurements are collected with an emphasis on collection during weather events, like rain and wind, during which we expect our classifiers to perform the worst. We demonstrate that the proposed classifiers outperform four other baseline classifiers in terms of false alarm probability by 1 to 4 orders of magnitude, and in terms of the misclassification probability

by 1 to 2 orders of magnitude. [1]

4.2 Introduction

In many situations, it is important to know when a person or other object crosses over a border or boundary in an outdoor environment. Wildlife preservation agencies wish to know when poachers have illegally crossed into a preservation area; national agencies wish to curb drug trafficking across national borders; farmers want to prevent loss of crop to thieves [1]. In many cases, the economic cost is too high to hire personnel to patrol the boundary of interest. In this paper, we propose and test an automated system of wireless sensors that detects and localizes when and where a boundary crossing occurs.

The idea of detecting and locating a tag-free person in sensor networks has already been developed and tested using several device-free localization (DFL) technologies. Using radio frequency (RF) links as the sensing method has been increasingly studied because of the ability to sense through walls and other objects opaque to light. Radio tomographic imaging [2], for example, measures received signal strength (RSS) on many wireless RF links and can achieve sub-meter localization errors in indoor environments. In this paper, we develop new DFL methods that are built specifically for the outdoor border crossing localization problem and we show that our methods can outperform current DFL methods.

One of the features of a boundary is that it is typically made up of a series of piecewise linear sections that divide two regions. A natural way of covering the boundary is to match the sensing system's geometry with that of the boundary: straight line segments. To this end, we build on the concept of using a wireless RF link as a line segment sensor [3], [4] where, if a person crosses the imaginary line segment between a transmitter and receiver, i.e. "link line," an algorithm can detect the crossing based on changes in the RSS measurements of the wireless channel.

Border sensing is notoriously challenging since sensors are prone to false alarms and

[1] In this joint work, I was a part of designing and implementing process of the system for boundary crossing detection system using the TI CC2531 and a custom energy harvesting system for long-term outdoor deployment. This section of the chapter appears in IPSN '16: Proceedings of the 15th International Conference on Information Processing in Sensor Networks, "Highly Reliable Signal Strength-Based Boundary Crossing Localization in Outdoor Time-Varying Environments", Peter Hillyard, Anh Luong, and Neal Patwari.

missed detections, which have serious consequences [5]. In this paper, we propose to deploy wireless nodes linearly along a boundary, partitioning the boundary into multiple short segments between neighboring nodes. The mesh network the nodes form creates many overlapping RF links. Limited in accuracy by themselves, the RF links, when overlapped in this deployment geometry, provide redundancy to improve crossing detection reliability and better localize which partition of the boundary is crossed [6].

We develop two new classifiers which use noisy RSS data from RF links in the network to determine the state of the boundary, either vacant or being crossed, and if being crossed, between which two nodes. This work contributes in several ways to the state-of-the-art:

- The new classifiers use soft decisions on each link, rather than hard decisions [6], and we show these allow significantly improved performance.
- Our classifiers do not use any labelled training. Rather, we develop simple statistical models whose parameters are estimated from a 5 second calibration period when it is known that the boundary is not being crossed. These parameters are re-estimated periodically during operation to adjust for time-varying changes in RSS measurements like those caused by weather events [7]–[9].
- We build and test a system running one of the classifiers in real-time for a three month outdoor deployment. We also recorded and make available over 77 hours of RSS data from the network [10], predominantly during weather events like rainstorms and wind when we anticipated the boundary monitoring would perform the worst.
- We use the data to compare the proposed classifiers with four baseline classifiers and show that the proposed classifiers can reduce the false alarm rate by 1 to 4 orders of magnitude or the misclassification rate by 1 to 2 orders of magnitude.

The remainder of this paper is organized as follows. Section 4.3 describes the overall boundary crossing localization system, the two proposed classifiers, and the adaptive RSS models we develop. Section 4.4 outlines the equipment used, the data collection experiments performed, and the validation metrics used to evaluate the proposed classifiers.

Section 4.5 reports the results of the experiments. We describe related research in Section 4.6 and conclude the paper in Section 4.5.

4.3 Methods

4.3.1 Boundary Crossing Localization System

In our border crossing localization system, N nodes are deployed linearly along a border (see Figure 4.1).

The nodes provide a natural series of line segments between neighbouring nodes that partition the boundary. We call the line segment between node j and $j+1$ "short segment j." The system we propose performs boundary crossing localization by classifying which short segment j a person is crossing. At any given time, our system outputs $\hat{j}$ where $\hat{j} \in \{0, 1, \ldots, N-1\}$. When $\hat{j} = 0$, our system indicates that no one is crossing the boundary at any location. In this wireless network, RF links form between each pair of nodes. For each of these links, the RSS, also called the received power in decibel units, is measured as r_i for wireless link i. The RSS is typically a discrete-valued measurement, and we denote its possible values as $\mathcal{S}$. We note that $\mathcal{S}$ also includes $\oslash$, the event that there was a missed packet and as such RSS was not measured. We observe a vector $\mathbf{r} = [r_1, r_2, \ldots, r_L]$ on L links.

Our boundary crossing localization system also consists of a model parameter update module and a crossing segment classifier (see Figure 4.2). The vector $\mathbf{r}$ is passed to the crossing segment classifier where we classify which section of the boundary the person crossed. The vector $\mathbf{r}$ is also passed to the model parameter update module where a buffer of recent RSS samplesIn an outdoor environment, we cannot count on the measurement statistics to stay the same over time. To address this reality, measurement statistics are periodically passed to the crossing segment classifier to update the models to account for time-varying changes in the RF channels. Later sections give further explanations of the update module and classifier.

4.3.2 Link Line Obstruction Model

In our proposed crossing segment classifiers, RSS measurements are used as evidence of a person being on or off of the link lines in the wireless network. Several statistical mod-

els currently exist that map RSS values or changes in RSS to probabilities of a person being on or off a link line [11]–[13]. The models in [11], [13], built using indoor measurements, can be used if the node locations are known. The model in [12] is independent of node locations, but there are eight tunable parameters. The models that we develop, which have three tunable parameters and are independent of node locations, are initialized and updated by using the statistics of the RSS measurements. Our models allow us to quickly and easily adapt to time-varying outdoor environments caused by rain and humidity [9], wind [7], and temperature [8].

Our model is based on the probability distribution of RSS measurements when a person moves along any point of the link line, and when a person moves somewhere away from the link line. We conduct a short experiment to demonstrate the differences in the distributions for these two cases. A person first walks along a link line, back and forth between the two nodes, for several seconds. The person then walks 6 ft away from and parallel to the link line for several seconds. We show the histograms of a link's RSS in Figure 4.3. We observe that when a person is off the link line, the RSS measurements tend to have very little variance. When no objects move near the link line, multipath are not likely to change, and thus, RSS remain relatively constant. However, when a person is on the link line, we observe large variations in RSS. The distribution has a larger variance and the mean value has shifted.

We use these observations to model the on and off link line distribution of RSS in dB as Gaussian, but modified to account for missed packets $r_i = \oslash$ and for numerical stability. We define the "off link line" probability mass function (pmf) for link i as,

$$f(r_i \mid \text{Off}) = \begin{cases} \rho, & r_i = \oslash \\ \frac{1}{\gamma} \max\left\{\epsilon, \mathcal{N}(r_i; \nu_i, \sigma_i^2)\right\}, & r_i \neq \oslash \end{cases} \tag{4.1}$$

where r_i is the RSS measurement, ρ is the probability of a missed packet, $\mathcal{N}(r_i; \nu_i, \sigma_i^2)$ is the normal function evaluated at r_i for mean ν_i and variance σ_i^2, $\epsilon > 0$ is a small-valued lower bound on the probability value away from zero, and γ is a constant such that the sum of the pmf equals one. The use of the minimum probability ϵ is due to the fact that in practice, we may observe values far from the mean more often than described by (4.1) and (4.2) because temporal fading does not always fit the log-normal distribution [14]. Using a small value ϵ conveys the model uncertainty and avoids numerical issues with very low

probabilities in the proposed classifiers. Unless otherwise stated, we let $\epsilon = 0.0001$, and $\rho = 0.03$.

To estimate the mean and variance of link i's off pmf is not known a priori, we estimate these parameters from a 5 second calibration period in which it is known that the boundary is not being crossed. To estimate ν_i, we use the median of the RSS measurements during calibration since the sample median is more robust to outliers than the sample average. To estimate σ_i^2, we use the maximum of the sample variance of the RSS measurements during calibration and a minimum constant $\omega^2 > 0$. Due to quantization of RSS, the sample variance may be zero even though the true real-valued received power would have had a positive variance. We impose a minimum variance of $\omega^2 > 0$ to avoid numerical instability. Unless otherwise stated, we let $\omega = 0.75$(dB).

Since our classifiers do not use labelled training data, we do not have access to RSS measurements when a person is crossing the boundary and thus have no way of knowing the pmf of RSS during a link line crossing. In outdoor deployments where line-of-sight exists, our experience is that we are likely to see a decrease in RSS. Thus, in an effort to make the model simple to compute, we set the "on link line" pmf for link i as,

$$f(r_i \mid \text{On}) = \begin{cases} \rho, & r_i = \oslash \\ \frac{1}{\gamma} \max\left\{\epsilon, \mathcal{N}(r_i; \nu_i - \Delta, \eta\sigma_i^2)\right\}, & r_i \neq \oslash \end{cases} \tag{4.2}$$

where Δ is the average decrease in RSS on a link due to a person's presence, and $\eta > 0$ is a factor by which the variance is increased due to a moving person on the link line. We use the same estimate for ν_i and σ_i^2 as we used for link i's off pmf, and unless otherwise stated, we let $\Delta = 5.0$ dB and $\eta = 4.0$.

4.3.3 Maximum Likelihood Classifier

In this section, we describe our first proposed classifier which we call the maximum likelihood classifier (MLC). Given that a person is either crossing one of the short segments j or not crossing the boundary at all ($j = 0$), we know for each link i whether or not the RSS should be in the on or off link line state. We map the relationship between crossing short segment j and crossing link line i as a *codeword*. The codeword for a person crossing short segment j is $\mathbf{w}(j) = [w_1(j), w_2(j), \ldots, w_L(j)]^T$ where $w_i(j) = 1$ if by crossing j, the person also crosses link i. Formally,

$$w_i(j) = \begin{cases} 1, & \text{if } (l_i \leq j) \text{ and } (j+1 \leq k_i) \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

where k_i and l_i are the endpoints of link i, and $k_i < l_i$, without loss of generality.

From the codeword, we can decide if r_i is a sample from $f(r_i \mid \text{Off})$ or $f(r_i \mid \text{On})$. The probability of observing r_i given short segment j is crossed can be computed as $b_j(r_i) = f(r_i \mid \text{On})^{w_i(j)} f(r_i \mid \text{Off})^{1-w_i(j)}$. Since $w_i(j) \in \{0, 1\}$, we pick off the probability of observing r_i given the crossed short segment. By applying independence between RSS measurements on the links, we obtain the likelihood of observing $\mathbf{r}$ by

$$b_j(\mathbf{r}) = \prod_{i=1}^{L} f(r_i \mid \text{On})^{w_i(j)} f(r_i \mid \text{Off})^{1-w_i(j)}. \tag{4.4}$$

The MLC decides which short segment j was crossed as,

$$\hat{j} = \arg\max_{0 \leq j \leq N-1} b_j(\mathbf{r}). \tag{4.5}$$

4.3.4 Hidden Markov Model Classifier

In this section, we describe a hidden Markov model (HMM) classifier that decides what short segment j was crossed. The HMM classifier (HMMC) is based on the forward algorithm [15]. We say that the boundary can be in one of N states at any given time: an "off" state which is when no short segment is being crossed; or a state that short segment j is currently being crossed. We denote the off state as S_0 and crossing-short-segment-j state as S_j. The probability that the HMM begins in S_j is denoted $\boldsymbol{\pi} = \{\pi_j\}$ where

$$\pi_j = P[q_1 = S_j], \quad 0 \leq j \leq N-1 \tag{4.6}$$

and q_1 is the state of the HMM for the first RSS measurement. In this paper, we let $\pi_0 = 0.9$ and $\pi_j = \frac{1}{N-1}(1 - \pi_0)$ for $j \neq 0$. The one-step transition probability from state i to state j, a_{ij}, is defined as

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i], \quad 0 \leq i, j \leq N-1. \tag{4.7}$$

In this paper, we assume that the boundary will be crossed infrequently and so the HMM will likely stay in S_0. Under this assumption, we let $a_{00} = 0.75$ and $a_{0j} = \frac{1}{N-1}(1 - a_{00})$ for $j \neq 0$. When the boundary is crossed, we assume that the person moves off the boundary quickly. As such, we let the probability $a_{jj} = 0.6$ for $j \neq 0$ and $a_{j0} = 1 - a_{jj}$ for $j \neq 0$. All

other entries not specified are 0 since a boundary crosser will only cross one short segment at a time. The HMM uses (4.4) to describe the probability of observing $\mathbf{r}$ given state j.

With this model, we are interested in solving the problem of: what is the most likely current state given a history of past $\mathbf{r}$ observations? The forward solution to the HMM answers this question with computational efficiency [16], inductively computing a vector $\alpha_t(j)$ at each time t for each state j, and then estimating the crossed short segment as $\hat{j} = \arg\max_{0 \le j \le N-1} \alpha_{t+1}(j)$. The forward algorithm initializes $\alpha_1(j) = \pi_j b_j(\mathbf{r}_1)$, and then computes $\alpha_{t+1}(j) = \left[\sum_{i=0}^{N-1} \alpha_t(i) a_{ij}\right] b_j(\mathbf{r}_{t+1})$ for each $t > 0$ and for $0 \le j \le N-1$.

4.3.5 On-the-fly Model Update

For wireless RF links, changes in temperature, humidity and movement of foliage during windy periods can cause the distributions of RSS to change compared to those taken at a prior time [7]–[9]. An example of changes caused by heavy rainfall can be seen from our own data collection in Figure 4.4.

We observe that as the rain starts, the RSS on this link drops steadily by approximately 8 dB in five minutes. After the rain stops, the RSS on the link increases, over two hours, back to approximately the same value prior to the start of the rain. This happens because, as the rain collects on the surface of the soil, the electromagnetic properties of the soil also change [17]. The phases and amplitudes of any reflection from the soil change, and thus the RSS changes as a result. As the rain abates, the water gradually evaporates or is drawn into the soil and over the course of hours, the electromagnetic properties of the soil are restored to the pre-rain state.

Gone unchecked, these time-varying changes to RSS caused by weather conditions would degrade the performance of the classifiers. To address these realities, we periodically update each link's on and off pmf using a history of RSS measurement. However, the history of measurements only include those that are observed when it is likely that no one is crossing the boundary, i.e. $\hat{j} = 0$. To accomplish this, we add the current RSS measurement r_i into a M-length FIFO buffer when $b_0(\mathbf{r}) > \kappa$ for the MLC and $\alpha_{t+1}(0) > \kappa$ for the HMMC. After β new RSS measurements are added into the buffer, ν_i and σ_i^2 are re-estimated for each link i using the measurements from their buffers. The re-estimated parameters are then used to update the off and on pmfs for each link simultaneously. In

this paper, we set $\kappa = 0.6$, $M = 50$, and $\beta = 18$. The buffer size allows us to update our models approximately every 5 seconds. Our model-update can quickly adapt to RSS changes caused by weather events because the rate of change of RSS during a weather event, like rain, is typically hundreds of times slower than the change caused by a person crossing.

4.3.6 Baseline Classifiers

In this section, we briefly describe four baseline classifiers against which we compare our proposed classifiers. The first two classifiers, from [6], make use of a binary vector that is produced by link line crossing detectors that determine the crossed-state of each link. The closest codeword classifier (CCC) finds the closest match, in Hamming distance, between the binary vector measurement and the same codewords referenced in Section 4.3.3. The short segment j whose codeword is closest in Hamming distance to the observed binary vector is used as the classified short segment crossed. Unlike the CCC, the simple classifier (SC) operates on a network of nodes where links form only between nodes j and $j+1$, for $j \in \{1, \ldots, N-1\}$. The SC decides short segment j is crossed only when its associated link measures a crossing. The SC ignores the measurements on longer links thus relying only on the individual shorter links for classification.

The third classifier is based on radio tomographic imaging (RTI) [2] and we refer to it as the radio tomographic imaging classifier (RTIC). The objective of RTI is to estimate an image of RF signal attenuation using RSS measurements on a network of RF links. The implementation of RTI in this paper uses the difference between a short and long term RSS average as the measurement vector from which we estimate the image. The image is estimated for each new vector measurement and the pixel with the greatest value is chosen as the location of the person. However, if the maximum pixel value is below a threshold, the boundary is assumed to be vacant. When RTI is applied to the boundary crossing localization system in this paper, the image we wish to estimate is one row of pixels that extends from node 1 to node N. Each pixel maps to one short segment, thus the RTIC maps the location estimate from RTI to the short segment j crossed. When the maximum pixel value is below a threshold, RTIC decides $\hat{j} = 0$.

The last classifier is based on the fingerprint-based system, *Nuzzer* [18]. In *Nuzzer*,

training RSS measurements are recorded as a person moves around in several distinct locations. Histograms of the RSS for each link are saved in a database for each of the areas in which the person is moving. In the *Nuzzer* classifier (NC) used in this paper, RSS measurements are recorded when a person stands far from the boundary, when a person walks along each short segment j, and when a person walks 1.21 m away from and parallel to each short segment. As explained in [18], the likelihood of $\mathbf{r}$ collected during testing is computed for each location using the probabilities from the training histograms. Smoothing is applied by multiplying a number of consecutive likelihoods. The NC decides $\hat{j} = 0$ if the maximum "smoothed" likelihood maps to any of the locations where the person was far from the boundary or if they were walking parallel to a short segment. Otherwise, the short segment with the maximum smoothed likelihood is classified as the crossed short segment. We apply a similar update method to NC as we designed for the MLC and HMMC in Section 4.3.5. In this update method, $\mathbf{r}$ is added to a buffer when $\hat{j} = 0$. When β new measurements have been added to the buffer, the histograms associated with a person far from the boundary are updated.

4.4 Experimentation

In this section, we describe the equipment, environment, experimental setup and metrics we use to quantify the performance of the proposed and baseline classifiers.

4.4.1 Equipment and Setup

Our nodes are Texas Instrument CC2531 USB dongles which follow a TDMA protocol and transmit +4.5 dBm power in the 2.4 GHz ISM band. An extra node overhears the wireless traffic and records $\mathbf{r}$ on a computer. On 1 meter tall stands, six nodes are deployed 4.6 m apart, making the total border length 23 m long. We measure RSS on all links and on four channels every 100 ms. Although we measure on four channels, it requires additional transmit energy and bandwidth compared to one channel. As such, in this paper, we analyze the performance of the classifiers when RSS measurements on one or on all four channels are available.

Using different deployments, we show how one of the proposed classifiers performs in a real-time system. We also demonstrate how the performance of the proposed classifiers

compare to the baseline classifiers in post-processing. One contribution of this paper is that we test the HMMC in a real-time system over a three month period. We build and deploy a system in which each node is powered by solar-rechargeable batteries. Additionally, a solar-powered BeagleBone Black is used to record RSS measurements and run the HMMC to perform real-time boundary crossing localization. We envision that the result could be sent to a server and used to remotely monitor or dispatch someone to the boundary; our deployed system simply captures and saves a webcam image along with the date, time, and estimated crossing location, to file [19]. We use these photos to establish the ground truth.

The HMMC in our deployment operates on the RSS measurements from all four channels, using 75% more power than when it measures on only one channel. However, we ran our three month deployment using four channels because it was within the capacity of our solar-rechargeable batteries, and we wanted to achieve the best performance possible.

In addition to the real-time setup, we also show how the performance would be affected in the real-time system if other classifiers or classifier parameters had been used. To that end, we perform many experiments where we record RSS measurements to file and run our proposed and baseline classifiers on those data sets in post-processing. In this paper, nodes are deployed in a variety of locations and weather conditions. The RSS measurements on all four channels are recorded in these cases. Note we evaluate the classifiers using data from either one or all four channels in Section 4.5.2.1.

4.4.2 Experiments

4.4.2.1 Experiments for postprocessing

To compare the performance of our proposed and baseline classifiers, we conduct a total of nine data collection experiments over the course of a month and a half at three different locations shown in Figure 4.5.

We refer to these locations as the field, the school, and the natural area. There are two large deciduous trees that stand within 6.1 meters of the nodes at the field, and five deciduous trees that stand within 1.5 meters of the nodes at the school. However the natural area is marked with high grasses, low shrubs, and deciduous seedlings and trees. Seven of the nine data collection experiments are conducted in the field. In three of these

seven experiments, measurements are recorded on sunny, calm days. In the other four experiments, we record several hours of RSS before, during, and after light to heavy rainfall. The experiments at the school and at the natural area are performed during times of wind with gusts up to 5.4 m/s measured by an anemometer. Since weather events, such as rain, cause time-varying changes in RSS, we expect that classifiers would perform the worst during these times. As a result, we purposefully chose to over-represent periods of rain so that we could learn more about the system when classification was most challenging. In all, 77.6 hours worth of RSS measurements are recorded over the nine experiments.

For each of the nine experiments, a person crosses the boundary at least 100 times and he or she records the crossing time and the short segment crossed. In most cases, ten seconds separate each of the crossings, although as much as 60 s elapse between some crossings to ensure the crossing times are properly logged. Data is then recorded for a time period when no one crosses the boundary, but when there may be activity near the boundary. An experiment lasts anywhere from one to fifteen hours. During this time, a rain meter and/or anemometer are used to collect the amount of rainfall and the wind speed near the deployment. We use these data sets to analyze the performance of each classifier during different weather events.

4.4.2.2 Experiments for real-time system

Two types of experiments are performed in the real-time system, which is deployed for three months in the field location. As the real-time system runs, we perform controlled tests where we record the time and location of 2887 boundary crossings. We then compare the true crossing times with the images saved to file on the BeagleBone Black. The second type of experiment we perform captures images during detected boundary crossings without a person purposefully crossing the boundary. The field is in the backyard of a residence, and some of these crossings (as shown by the captured images) are due to residents or animals crossing, and some are false alarms.

4.4.3 Validation Metrics

Each classifier may produce false alarms or misclassify the true crossed short segment. A false alarm occurs when, at any observation, the boundary is vacant, i.e. $j = 0$, but the

classifier reports $\hat{j} \neq 0$. We compute the probability of false alarm, P_{fa}, by counting the number of false alarms and dividing it by the total number of samples measured when no person is crossing the boundary. Next, a classifier makes a correct classification $\hat{j} = j$ for $j \neq 0$. In the experiments, we record the ground truth time of crossing and the short segment crossed. Due to small errors in the recorded true time of crossing and many samples worth of delay in the baseline classifier algorithms, a classifier may report the correct j but at a delayed time compared to the true crossing. To be fair to all classifiers, a correct classification in our paper occurs when $\hat{j} = j$ for $j \neq 0$ at any time during a ± 3 second time window. The probability of correct classification, P_{cc}, is thus the number correct classifications divided by the number of boundary crossings. We show an example of the HMMC output for a short test experiment in Figure 4.6.

In this short experiment, the HMMC achieves a $P_{cc} = 10/10$ classification rate and produces no false alarms, $P_{fa} = 0$.

None of the classifiers we implement are able to achieve the ideal of $P_{cc} = 1$ and $P_{fa} = 0$ over all experiments. Instead, compromises must be made so that a classifier's tunable parameters are selected to penalize one type of error more than the other. For example, in some scenarios, missing a classification may cost €1000 in stolen goods while a false alarm may only cost €100 to have a worker turn on a video-surveillance system. In a different scenario, a false alarm may waste €1000 in man-power and fuel to investigate a distant boundary, while a misclassification may only cost €100. To quantify the tradeoff between the probability of misclassification, $1 - P_{cc}$, and P_{fa}, we define a cost function C_k:

$$C_k = c_{k0} \cdot P_{fa} + c_{k1} \cdot (1 - P_{cc}) \tag{4.8}$$

where k is the cost function number. The choice of c_{k0} and c_{k1} quantify the cost of each type of error.

4.5 Results

4.5.1 Sensitivity to Parameter Choice

Our classifiers rely on the selection of three tunable parameters. One important feature of our work is that few parameters need to be adjusted to maintain a high performance. With few parameters to adjust, this system can be deployed quickly. In this section,

we show how the choice of the parameters affects the performance metrics described in Section 4.4.3. We evaluate performance using the data in post-processing by averaging over all seven data collection experiments, and over all four channels, but using one channel at a time.

4.5.1.1 Link line model parameters

Section 4.3.2 describes the Gaussian models we use for the distributions of RSS when a person is off or on a link line. In this section, we vary the model's tunable parameters Δ, η, and ω. We observe in Figure 4.7 the affect of η and ω and $\Delta = 5$ on performance. In Figure 4.7, we observe a few trends that are common to both the MLC and the HMMC. First, when η increases, both P_{cc} and P_{fa} increase. The η parameter determines the scale of the on link line distribution. A higher η, and thus a wider "on link line" pmf, favors RSS observations coming from the on link line distribution for all links. As a result, we are more likely to correctly classify crossings. But we also increase the chance of misclassifying boundary crossing locations. This trade-off does not seem to be as significant with the choice of ω. When using either the classifier, the user must consider the cost of false alarms and misclassifications when choosing η. Although we have included figures for $\Delta = 5$, we found that Δ should remain between 3 and 7 for best results. For $\Delta < 3$, the on and off link line distributions are considerably overlapped and cause a low classification rate. For $\Delta > 7$, the on link line distribution falls too far away from the observations to reliably classify crossings.

Another important observation from Figure 4.7 is the effect of the classifier on performance. The MLC shows a higher variability in P_{fa} than for the HMMC. Over the parameters tested, the highest false alarm probability for HMMC is $P_{fa} = 7 \times 10^{-5}$ whereas MLC reaches $P_{fa} = 2.5 \times 10^{-4}$. We suspect that this difference is because the HMMC tends to smooth out crossing detections by placing a prior on the observation likelihood. This smoothing characteristic tends to reduce the number of false alarms. However, the smoothing characteristic on the HMMC also reduces the probability of correct classification compared to MLC. The HMMC achieves a maximum of $P_{cc} = 0.9968$ while the MLC achieves a maximum $P_{cc} = 0.9978$. The HMMC is, in general, better when the application asks for few false alarms, but the MLC is more accurate when the application demands a

high classification rate.

4.5.1.2 One-step transition probabilities

Both the MLC and the HMMC rely on the likelihood model (4.4) to perform classification. However, the HMMC also makes use of the one-step transition probabilities a_{ij} for $0 \leq i, j \leq N-1$. These are additional parameters that the user must select when implementing the HMMC, so we would like to know how performance responds to the choice of these transition probabilities. In Section 4.3.4, we described how the transition probabilities could be defined in terms of a_{00} and a_{jj} for $j \neq 0$. We loop over many values for these parameters and record P_{cc} and P_{fa} for the HMMC. We observe in Figure 4.7 the affect of a_{00} and a_{jj} on performance.

In Figure 4.7, we observe distinct features common to both P_{fa} and P_{cc}. First, the general trend shows that as a_{00} increases, P_{cc} decreases while P_{fa} increases. When a_{00} is small, we increase the probability that on the next RSS observation, the HMM will transition from S_0 to S_j for $j \neq 0$. The model favors leaving S_0. Thus, we observe a high P_{cc} and a high P_{fa}. However, when a_{00} is high, the model favors staying in S_0. Thus we observe fewer false alarms but also fewer correct classifications. The opposite trend is true for a_{jj}: P_{cc} and P_{fa} tend to increase as a_{jj} increases. A small a_{jj} favors transitions to S_0 if the HMM is currently in state S_j for $j \neq 0$. In this case, it is likely the HMMC will result in $\hat{j} = 0$ which yields a low P_{fa} and P_{cc}. However, when a_{jj} is large, the model favors staying in S_j for $j \neq 0$ on the next observation. As a result, we observe increases in both P_{cc} and P_{fa}.

One very distinct part of the relationship between P_{fa}, P_{cc}, and the transition probabilities is the contour between the relatively flat sections and steep sections. The flat regions suggest that when a_{00} and a_{jj} are appropriately chosen, the transition probabilities carry much less significance in the classification process. But poorly chosen transition probabilities can be significantly detrimental by causing a sharp increase in P_{fa} without a equally significant increase in P_{cc}. The take away from this analysis is to choose a_{00} and a_{jj} inside of the flat region to avoid costly increases in P_{fa}.

4.5.2 Classifier Comparison

In this section, we compare the performance of the proposed and baseline classifiers discussed in Section 4.3.6. We foresee two kinds of deployments in which one may implement any of these classifiers. In the first type of deployment, the system remains in the same location and the user has time to tune the parameters of the classifier. A person who wishes to monitor a relatively short boundary and never moves the nodes may fall in this category. In the second type of deployment, the user must quickly deploy the system in any type of environment, and there is no time to tune the classifier parameters. This situation may be common to a group who must protect miles of boundary and tuning parameters would require an inordinate amount of time. In either case, we are interested in implementing the classifier with the best performance.

In the following sections, we show the performance of all six classifiers when we are able to tune their parameters. Then, using the parameters that provide an optimal performance, we show how these classifiers perform when the system encounters a variety of weather conditions and when it is moved to a new location. By doing so, we analyze the performance of each classifier when no parameter tuning is performed.

4.5.2.1 Performance with tuning

We begin by running the six classifiers on the data from the seven field experiments on each of the four channels individually. For each classifier, we loop through several combinations of parameter values and record the average P_{cc} and P_{fa} over the four channels. We show the results of this process in Figure 4.8.

The ROC shows the relationship between P_{cc} and P_{fa} as the classifier parameters change. Even though none of the classifiers achieve the ideal of $P_{cc} = 1$ and $P_{fa} = 0$, some classifiers are able to approach this ideal better than others. The first observation is that NC performs the worst of the classifiers. The performance of NC suffers since the "on link line" training measurements collected on the first day gradually drift. The NC tends to have few false alarms since the histograms associated with $j = 0$ (off link line) are being updated. As new measurements are observed, the new measurements will tend to match the measurements from these updated histograms more than the histograms associated with a person moving on a short segment.

We also observe that the SC does not perform as well as the other classifiers. We suspect that this is because the SC does not incorporate the measurements from the longer links of the network for classification. Next, we compare the final 4 classifiers. The inset of the ROC in Figure 4.8 shows that RTIC, CCC, HMMC, and MLC all come close to achieving the ideal. However, MLC and HMMC appear to come closest to the ideal, followed by the CCC and then the RTIC. Although the CCC, which relies on binary measurements, outperforms the RTIC, it is outperformed by both MLC and HMMC. The proposed classifiers differ from the CCC in that they use soft values in classification whereas the CCC uses binary values.

To get a better sense of how these classifiers compare, we fix the classification rate to be $P_{cc} = 0.995$ and find the lowest false alarm rate for each classifier. We then fix the false alarm rate to be $P_{fa} = 5.0 \times 10^{-5}$ and find the lowest misclassification rate for each classifier. We show the results in Table 4.1. We observe that for a fixed P_{cc}, the MLC and the HMMC outperform the CCC by one order of magnitude and the SC by four orders of magnitude. For fixed P_{fa}, the MLC and the HMMC outperform the CCC and RTIC by an order of magnitude and the SC by two orders of magnitude.

We have only considered the case of measuring RSS on one channel. Now we analyze the performance of the MLC and the HMMC when RSS is measured on four channels. We again run the MLC and HMMC with many parameter combinations on the data from the seven field experiments. For many of these parameter combinations, the MLC and HMMC both achieve 100% classification of all 901 boundary crossings. The MLC and HMMC, in terms of misclassification, is therefore no greater than $1/901 = 1.11 \times 10^{-3}$. We show in Table 4.2 that, for a fixed $P_{fa} = 2.5 \times 10^{-5}$, the classifiers that operate on RSS measurements from four channels reduces the misclassification rate by at least one order of magnitude. Although the additional channels provide even better performance than for a single channel, this advantage must be weighed against the increased power usage to transmit on the additional channels. When power consumption must be minimized, measuring on one channel reduces the power usage by 75% compared to a network that measures on four channels.

4.5.2.2 Performance without tuning

In the previous section, we analyzed the performance of all six classifiers in the case when we were able to tune their parameters for a deployment in one location. In this section, we consider the case where the parameters for each classifier have already be optimized to meet some specification. We then test the performance of these classifiers in a variety of weather conditions and at different experimental locations. By doing this, we analyze how each classifier performs in a variety of environments when the parameters are fixed. Next we describe the data sets we use and how we choose optimal parameters for each classifier.

Different from the previous sections, we organize data-collection experiments by the type of weather condition: sun, rain, wind, and all-weather which are mentioned in Section 4.4.2. The all-weather group is the sun, rain, and wind data combined into one. Note the wind data is composed of data collected at the school and at the natural area. The sun and rain data, however, come from only the field location. For each weather type, we separately log the number of correct classifications, the number of true crossings, the number of false alarms, and the total number of RSS observations in the four types of weather conditions. We run all classifiers except the NC on each channel separately and then average the number of correct classifications and the number of false alarms for all four channels. We obtain P_{cc} and P_{fa} for each classifier in each weather condition. We note here that we do not consider the performance of the NC since we saw in the previous section that it is unable to achieve a high classification rate; furthermore, retraining such a classifier, in many situations, would be too costly.

Next, two sets of optimized parameters for each classifier are chosen through an optimization process. Using the P_{cc} and P_{fa} results from Section 4.5.2.1, we record the parameters that minimize two cost functions, C_0 and C_1, defined in (4.8). The cost function C_0 penalizes false alarms the most by setting $c_{00} = 1000$ and $c_{01} = 100$. The optimal parameters for each classifier with this cost function would be ideal for situations where the cost of investigating a boundary crossing costs time and resources. On the other hand, the cost function C_1 penalizes misclassifications the most by setting $c_{10} = 100$ and $c_{11} = 1000$. The optimal parameters for each classifier with this cost function would be ideal for situations where knowing the true crossing state is imperative while risking the

chance of some false alarms is allowable.

Using the optimized parameters, we run the classifiers on all four weather data sets and record P_{cc} and P_{fa}. We use these new P_{cc} and P_{fa} values to compute new cost values, C_0' and C_1', using the same c_{k0} and c_{k1}, for each classifier and each weather condition. We show these new costs of running each classifier on the data sets mentioned above in Figure 4.9.

In Figure 4.9, we observe, first, that the SC performs the worst in all weather conditions regardless of the optimized parameters used. The SC does not make use of all of the RSS measurements in this network configuration so it performs the worst. The best classifiers are the CCC, MLC, and HMMC when comparing the sun, rain, and all-weather data and in both optimized regimes. But in both optimized regimes, either the MLC or the HMMC perform the best. The numeric values for the CCC, MLC, and HMMC costs are shown in Table 4.3. The HMMC performs the best in these three weather conditions when the parameters are optimized to reduce false alarms. The HMMC showed this property in Section 4.5.1.1 where compared to the MLC, the HMMC was a better classifier for reducing false alarms. In contrast, the MLC outperforms the HMMC in the all-weather and rain conditions when we optimize parameters for correct classifications. The HMMC and the MLC are both comparable in the sunny weather. Again, this characteristic of the MLC appeared in Section 4.5.1.1 where compared to the HMMC, it was the superior classifier for correct classification.

We also see that the CCC is the second best classifier in many of the weather conditions and optimization regimes. This comes at little surprise in that the ROC curve in Figure 4.8 showed the CCC was only slightly less accurate than the HMMC and MLC. Perhaps the more surprising result is that the RTIC is the best classifier in windy locations and when optimizing for correct classifications. And, since the windy data sets were performed at two different locations, the RTIC performs the best out of the classifiers when the system is moved to new locations and when the classifier parameters are fixed. The MLC, however, is the best choice in windy conditions when we want to minimize false alarms.

We separately obtain cost values for the MLC and the HMMC when we disable the on-line model update feature. In both the C_0' and C_1' regime, the MLC and HMMC with the update feature enabled lowered the cost by 1 to 2 orders of magnitude with the sun

data and lowered the cost by 2 to 3 orders of magnitude with the rain data compared to these classifiers when the update feature was disabled.

From this analysis, we draw a few conclusions about our classifiers. First, the on-line model update feature is necessary to operate reliably during weather events. Second, when we take all weather conditions (sun, rain, and wind) into account, the MLC is the best classifier when correct classifications are important, but the HMMC is the best classifier when minimizing false alarms. However, it is advantageous to use the RTIC in windy conditions when correct classification is needed; the MLC, however, is the best choice in windy conditions when false alarms must be minimized.

4.5.3 Relative Costs

In this evaluation, we compare the costs of using the MLC and the HMMC by optimizing their parameters over many cost functions (4.8). In the previous section, we considered cases where the cost of a misclassification or a false alarm was ten times as costly as the other rate. However, there are an infinite number of cost parameters c_{k0} and c_{k1} a user may want to consider. To address this, we consider a relative cost C_{rel} which is a function of a ratio of the cost parameters where one of the cost parameters is fixed. In this section, we fix $c_{k0} = 1000$ and vary c_{k1} to be values between 1 and 100,000. Using the results from Section 4.5.2.1, we then obtain the parameters for the MLC and the HMMC that minimize C_{rel}. We then run the MLC and the HMMC using their optimized parameters on the all-weather data described in 4.5.2.2 to obtain values for P_{cc} and P_{fa}. These rates are then used to compute new relative costs C'_{rel}. We show the new relative costs in Figure 4.10.

We observe that when the cost of a misclassification is 2 or more times as expensive as the cost of a false alarm, the MLC can save more in cost compared to the HMMC. When the cost parameter ratio is in this regime, the classifier parameters are optimized to minimize misclassifications. When the cost parameter ratio, however, is less than 0.2, the HMMC generally reduces the overall cost. When the cost parameter ratio is in this regime, the parameters are optimized to minimize false alarms. Due to limited space we do not include plots, but the same general results are seen in the sun and all-weather data sets. In windy conditions, the MLC is the preferred classifier regardless of cost ratio.

4.5.4 Real-time Performance

In this final evaluation, we report the results of the real-time, three month deployment. We captured webcam images of when the system detected a boundary crossing during both controlled and uncontrolled periods. During the controlled experiments, a person crossed the boundary 2887 times. Of those crossings, 81.6% were correctly classified by the real-time HMMC. We note that the classification rate is 18% less than what is reported in the previous sections. Further investigation of the real-time system showed that approximately one out of six seconds worth of RSS measurements was not recorded by the BeagleBone Black. During the one second of missing measurements, the BeagleBone requests weather data from a website and saves it to file. We did not anticipate that the web request and saving would result in missed RSS measurements when we deployed the real-time system. As a result, 1/6 or 16.6% of the RSS measurements during a crossing were missing. This percent of missed measurements gives some explanation about the 18% lower classification rate between the real-time and post-processing results.

After changing the real-time system to refrain from requesting weather data, we run the real-time system for an additional five days, during which, a person crosses the boundary 1502 times. All 1502 crossings were correctly classified, which indicates that the probability of misclassification $1 - P_{cc}$ is less than $1/1502 = 6.7 \times 10^{-4}$. This classification performance matches values reported for post-processing in Section 4.5.2.1.

During the three month long uncontrolled deployment, the real-time system operated continuously during sunny, rainy, and windy weather. In all, 850 boundary crossing locations by people and animals were correctly classified. The true classification rate is unknown, however, because the total number of true crossings during the uncontrolled period is unknown. The system also reported 298 false alarms. Roughly 69 million RSS measurements were taken during this deployment, which results in a false alarm probability of 4.3×10^{-6}, or an average of 3.7 false alarms per day. We note that this rate is lower than the lowest 21.6 false alarms per day average achieved in post-processing. To reduce this rate, another row of nodes could be deployed to verify crossing detections. For example, the extra nodes could be placed at lower or higher heights or they could be deployed parallel to the first row of nodes. This adds yet another layer of redundancy to mitigate errors. We also found that false alarms were more likely to occur during weather

events. False alarms could be further reduced by ignoring RSS measurements that have the same signature as those cause by wind or rain.

4.6 Related Research

Outdoor boundary crossing localization using sensing systems has been a topic of research for many years. Since it cannot be assumed that the boundary crosser carries a tag that cooperates with the sensing system, boundary crossing localization typically revolves around DFL technologies. The sensing devices, however, vary considerably. Proximity sensors have been widely used. For example, significant research has been conducted using passive infrared (PIR) sensors for outdoor localization [20], [21] where the sensors detect thermal radiation from a person. These studies have investigated not only the localization algorithms [20], but also their energy efficiency [21]. A major disadvantage of PIR sensors is that they lose the ability to track moving objects in forested or vegetated landscapes where obstructions occlude the sensors' field of view. In addition, PIR sensors' performance suffers during daylight times when the infrared radiation from the sun saturates the sensor. Cameras have also been proposed as a sensing technology for boundary surveillance [22]. When boundaries are monitored day and night, cameras become useless at night or in fog when it is not possible to image and thus detect and localize boundary crossers. Moreover, cameras mounted on aerial units cannot image objects that are occluded from above, *e.g.*, by a forest canopy.

Fiber optic and pressure sensors have been used in boundary security applications to detect when a person's footsteps [23]. These sensors are either buried underneath the boundary or are strung along a fence. Installing buried sensors involves heavy equipment digging up earth. Maintenance also poses a problem since the cable must be unearthed for visual inspections and to perform repairs. Pressure sensors strung along the fence first requires the fence which is economically costly operation.

In contrast, RF signals, which we use in this paper, operate day and night, can pass through leaves and other vegetation, and can operate during fog and other weather conditions. The sensors are easily deployed, can be easily maintained, and do not require a fence to perform sensing. One type of RF sensing includes radar which has been used to detect human presence in wooded areas [24]–[26] and even distinguish human presence

from a soldier and a vehicle [27]. We note one well-reported commercial radar system used for boundary crossing localization produced false crossings during weather events and only worked 30% of the time [26], [28]. Radar uses a signal whose power decays with distance d as $1/d^4$, thus transmit power must be increased dramatically to increase sensing range. Our work closely matches other DFL methods that measure changes in the channel of RF links and whose power decays as $1/d^2$. These methods have been heavily tested in indoor environments.The work in [6] is optimized for linearly deployed nodes, but no testing is performed during weather events. Furthermore, [6] uses binary link line crossings. We show that methods using soft link line crossing measurements outperform localization methods that use hard crossing decisions. The methods we develop in this paper are optimized for linearly deployed sensors, and we show that they outperform current DFL technologies. Additionally, our model update feature allows us to achieve low error rates despite weather-induced propagation changes.

4.7 Conclusion

In this paper, we developed and tested two RSS-based crossing segment classifiers to localize boundary crossings. These classifiers used models whose parameters could be re-estimated from a history of RSS to adapt to changes caused by weather events. We evaluated our classifiers from experiments performed in three locations and in a variety of weather conditions. Using 77.6 hours worth of RSS measurements, we analyzed our classifiers in terms of sensitivity to system parameters. We found that the HMMC is better when the application asks for few false alarms, but the MLC performs better when the application demands a high classification rate.

When we compared the MLC and HMMC to four baseline classifiers, we found that the MLC and HMMC outperformed the other classifiers by 1 to 4 orders of magnitude in terms of false alarm probability. We also showed that the MLC and HMMC were the best classifiers in sunny, rainy, windy and in overall weather conditions when we fixed the classifier parameters to avoid false alarms. In addition, the MLC was more accurate than the HMMC when parameters were optimized to minimize misclassifications. Our three-month duration real-time six node boundary crossing system deployment achieves high reliability, with a misclassification rate of less than 7×10^{-4} and false alarm rate of

4×10^{-6}. In summary, using an MLC and HMMC algorithm on RSS data from a linearly-deployed network provides highly reliable boundary crossing localization across a variety of weather conditions and in multiple locations. We demonstrated that these classifiers can perform better than baseline DFL classifiers by tuning system parameters or using fixed, optimized parameters.

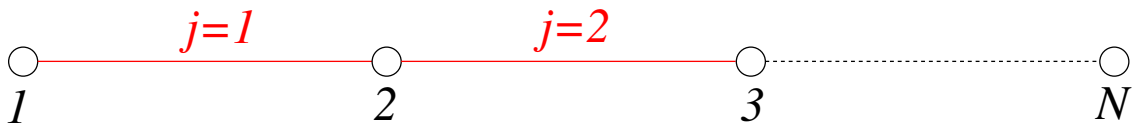


Figure 4.1. Nodes and short segments j in localization system.

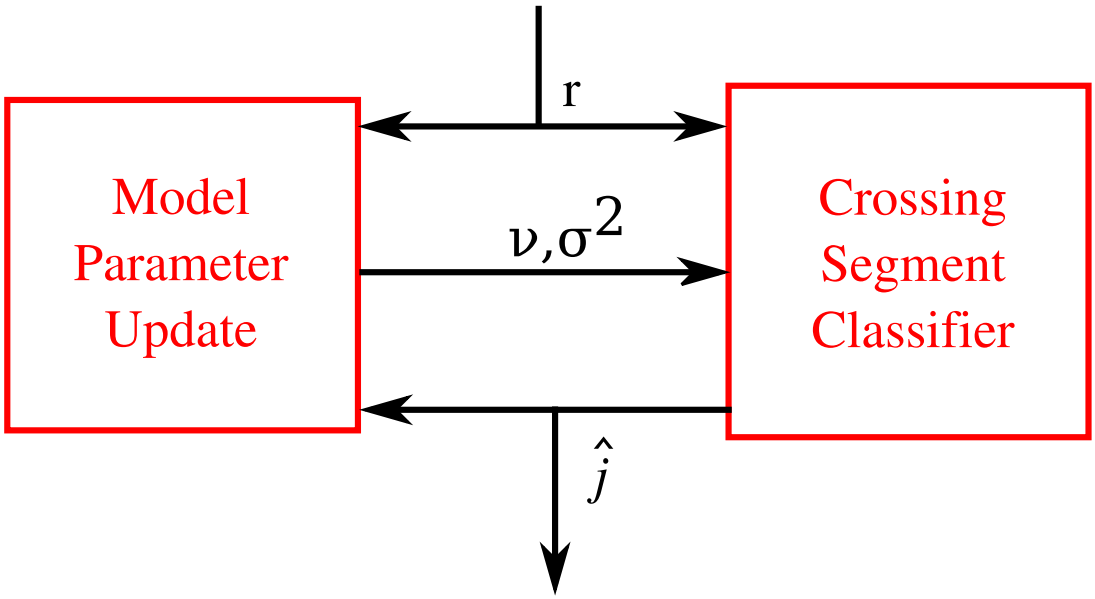


Figure 4.2. Block diagram of the proposed localization system.

Table 4.1. Classifier performance for fixed P_{cc} or P_{fa}

	MLC	HMMC	CCC	RTIC	SC
P_{fa} for fixed $P_{cc} = 0.995$	3.05×10^{-5}	4.11×10^{-5}	4.58×10^{-4}	7.10×10^{-2}	2.04×10^{-1}
$1 - P_{cc}$ for fixed $P_{fa} = 5.0 \times 10^{-5}$	3.33×10^{-3}	3.61×10^{-3}	5.80×10^{-2}	9.16×10^{-2}	$> 2.50 \times 10^{-1}$

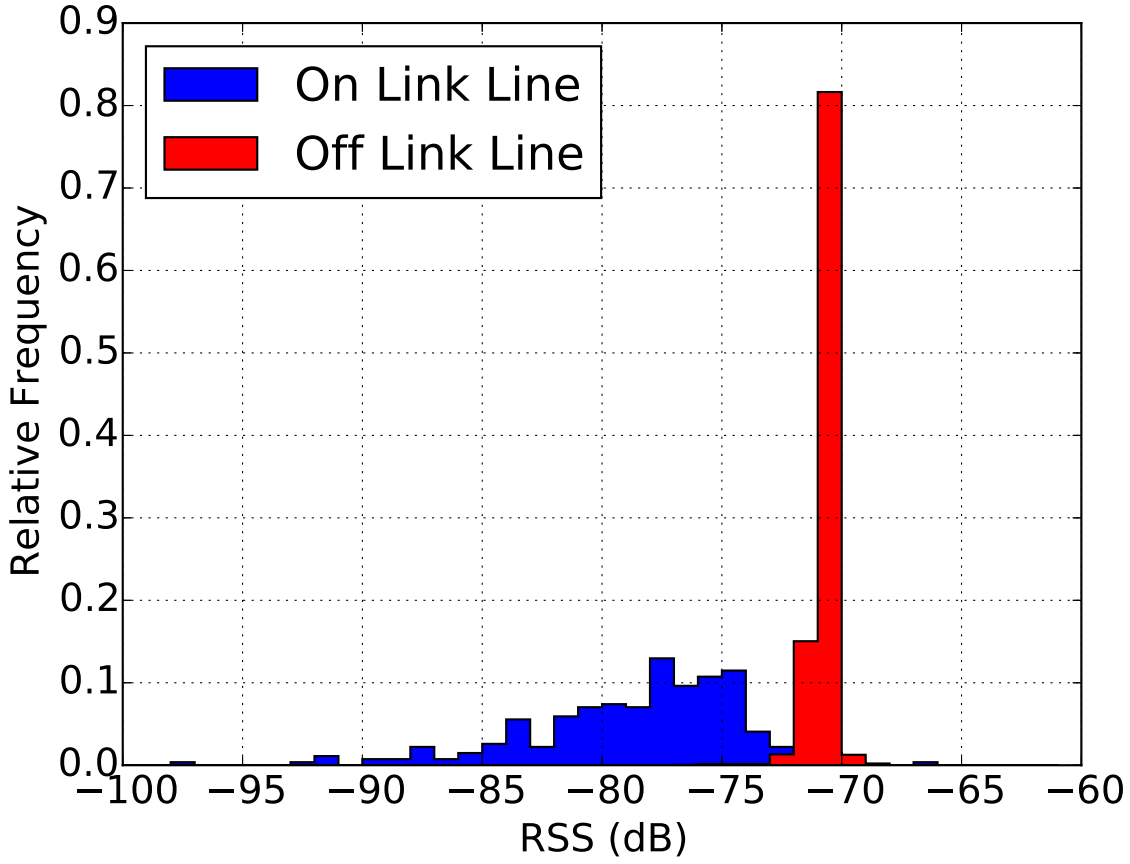


Figure 4.3. Relative frequencies of RSS measurements when a person is on, and off, a link line.

Table 4.2. Perf. vs. channels for fixed $P_{fa} = 2.5 \times 10^{-5}$

	MLC	HMMC
1 channel $1 - P_{cc}$	1.37×10^{-2}	2.80×10^{-2}
4 channel $1 - P_{cc}$	$< 1.11 \times 10^{-3}$	$< 1.11 \times 10^{-3}$

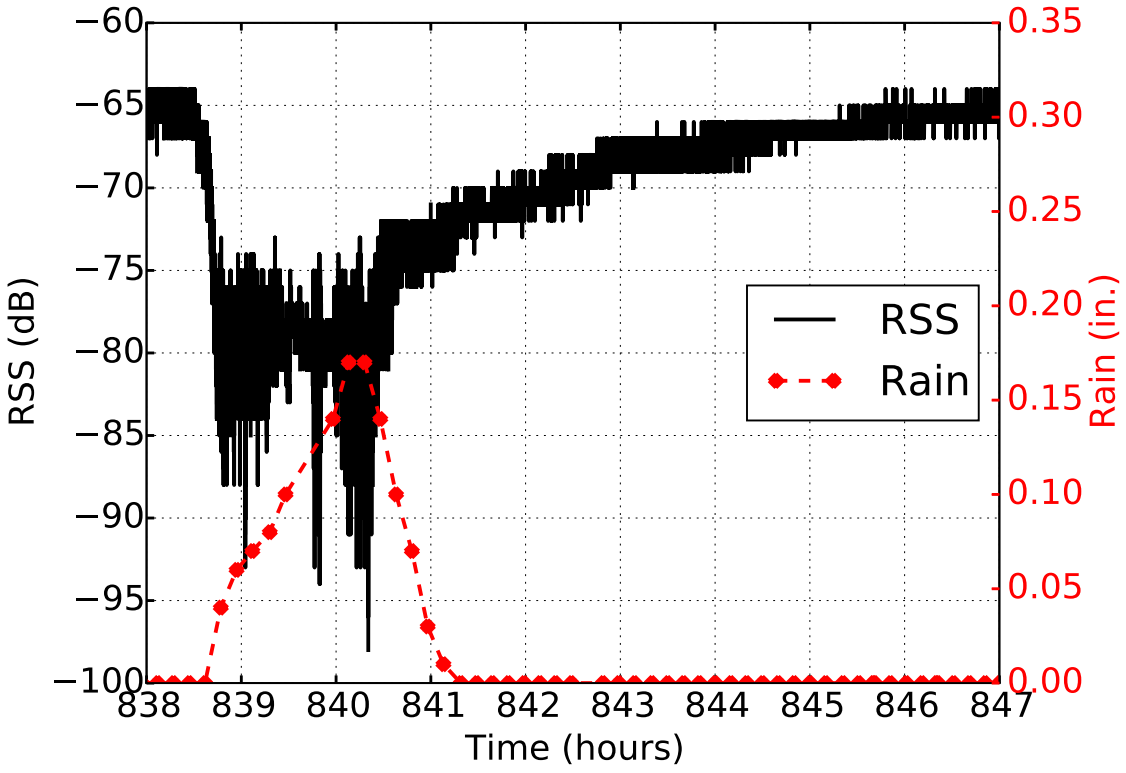


Figure 4.4. A link's RSS decreases over minutes during the start of rain and returns to the pre-rain level after the rain subsides.

(a) (b) (c)

Figure 4.5. Experiment locations: (left) Field (middle) School (right) Natural Area.

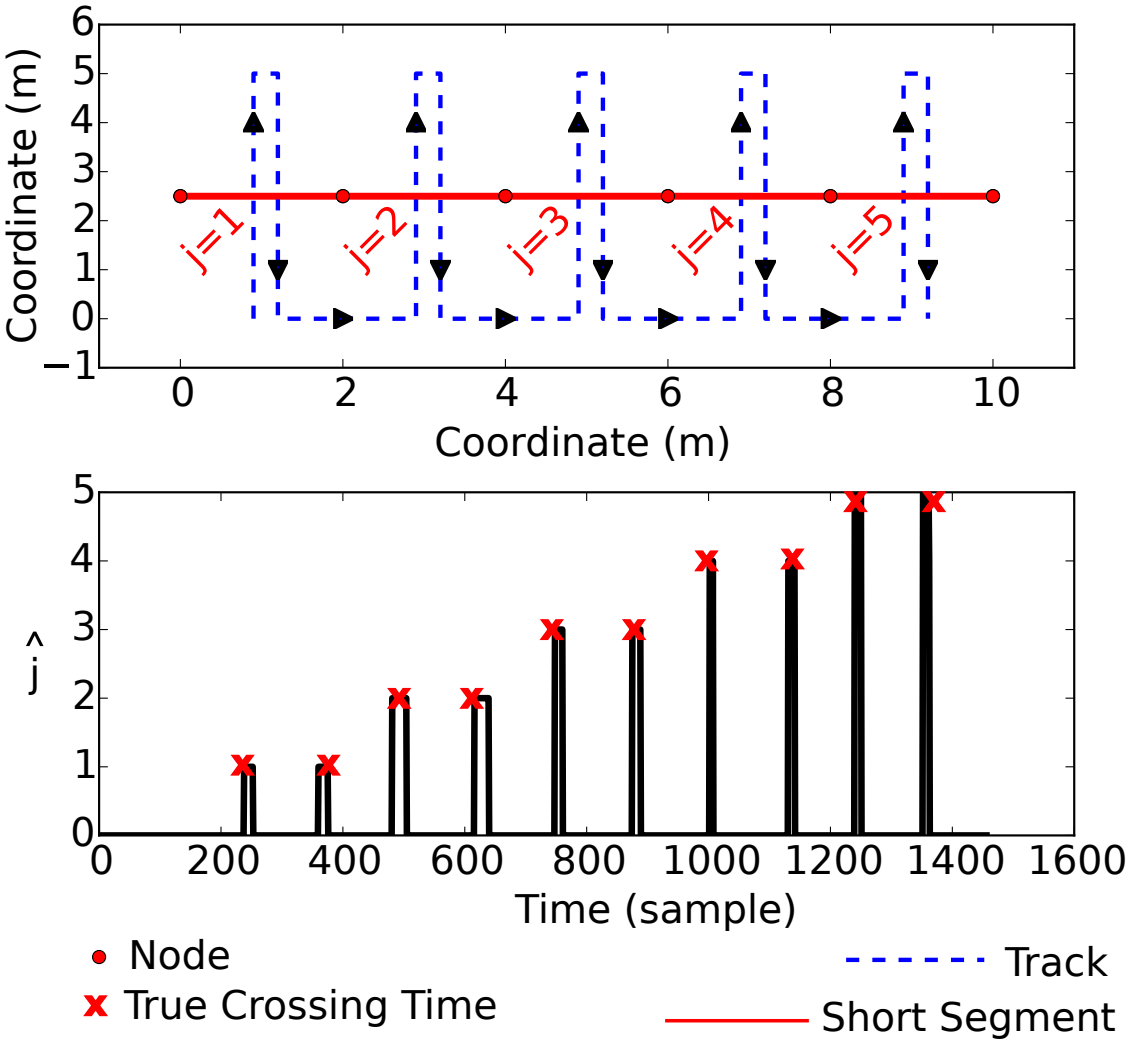
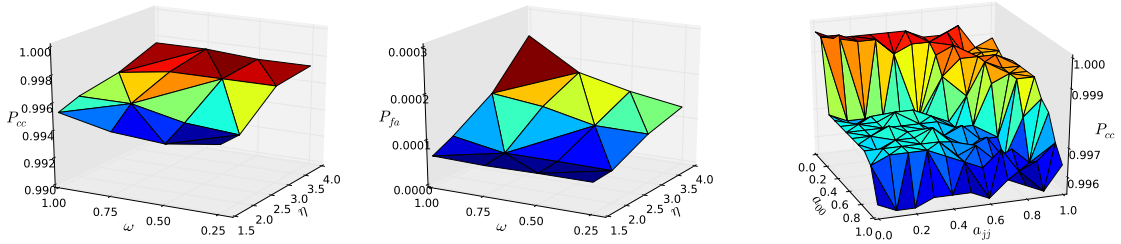


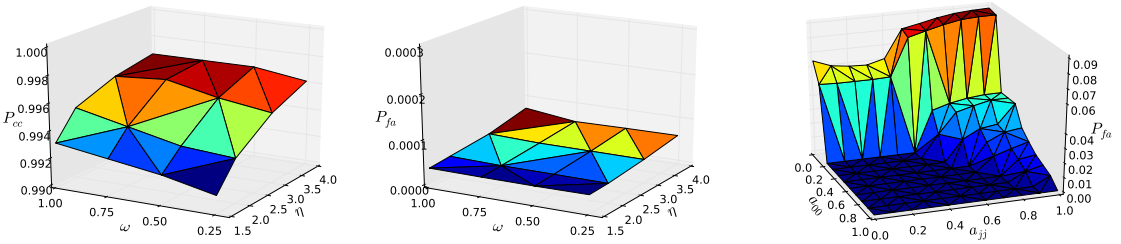
Figure 4.6. (above) The track of the person. (below) Output of HMMC for a short experiment.

Table 4.3. Numeric cost values

	C'_0			C'_1		
	All	Sun	Rain	All	Sun	Rain
MLC	1.93	0.56	0.79	3.04	2.69	0.83
HMMC	1.11	0.34	0.43	5.59	2.63	3.19
CCC	1.62	0.50	1.18	3.52	3.12	3.96

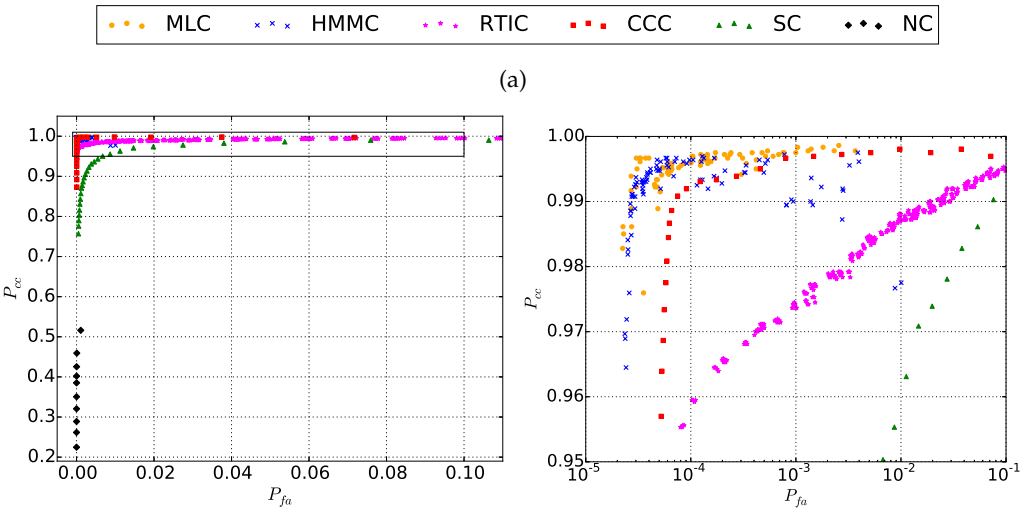


(a) P_{cc} for MLC (b) P_{fa} for MLC (c) P_{cc} for HMMC



(d) P_{cc} for HMMC (e) P_{fa} for HMMC (f) P_{fa} for HMMC

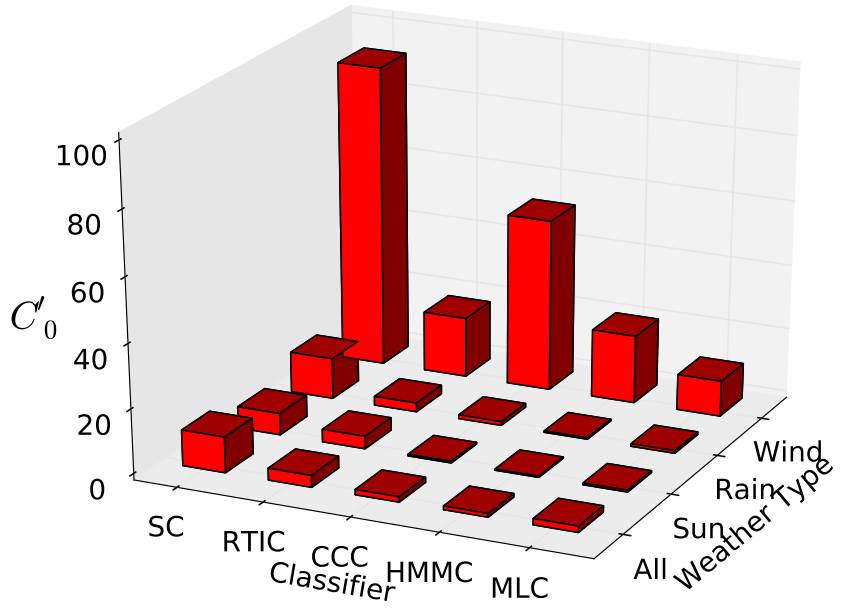
Figure 4.7. Effect of link line obstruction model parameters on MLC and HMMC (a)-(d); and one-step transition probabilities on HMMC performance (e)-(f).



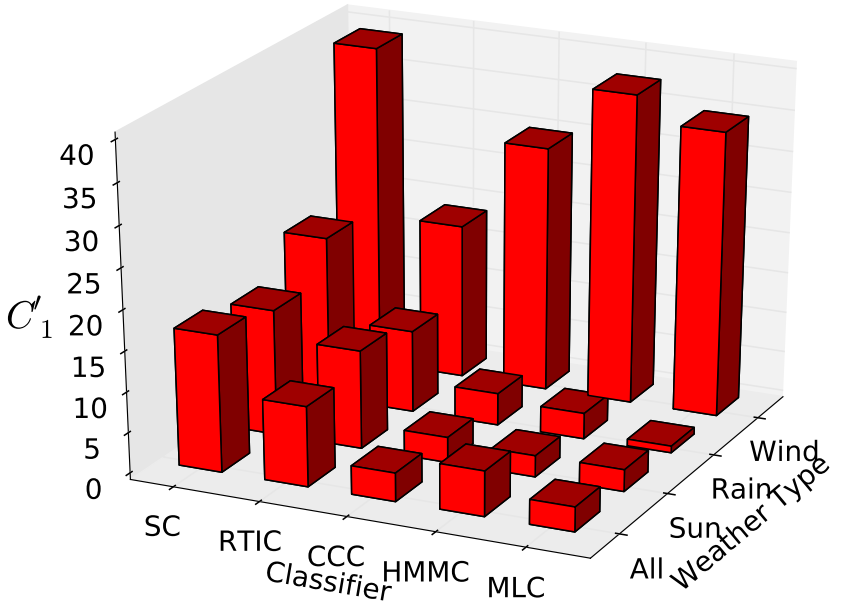
(a)

(b) (c)

Figure 4.8. (left) ROC curves for six classifiers. Inset box in top left. (right) Inset of the ROC curve.



(a)



(b)

Figure 4.9. (top) C'_0: parameters optimized to minimize false alarms. (bottom) C'_1: parameters optimized to minimize misclassifications.

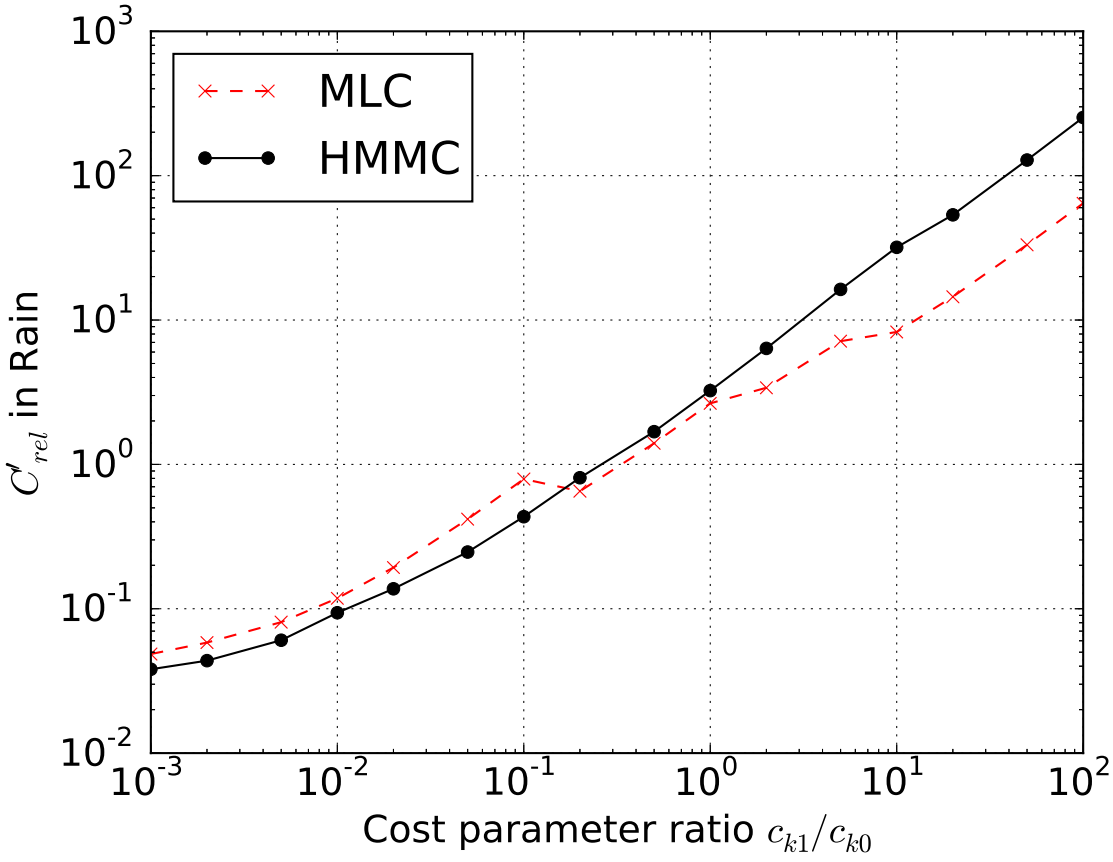


Figure 4.10. C'_{rel} for both the MLC and HMMC using the rain weather data sets.

4.8 References

[1] Melvyn Thomas, "Over 35,000kg mangoes stolen from orchards," 2015.

[2] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 5, pp. 621–632, May 2010.

[3] Kristen Woyach, Daniele Puccinelli, and Martin Haenggi, "Sensorless sensing in wireless networks: Implementation and measurements," in *2006 4th Int. Symp. Model. Optim. Mobile, Ad Hoc Wirel. Networks, WiOpt 2006*, 2006.

[4] Moustafa Youssef, Matthew Mah, and Ashok Agrawala, "Challenges: device-free passive localization for wireless environments," in *MobiCom '07: ACM Int'l Conf. Mobile Computing and Networking*, Montréal, Québec, Canada, Sept. 2007, pp. 222–229.

[5] Robert Beckhusen, "Homeland security delays plan to place sensors on U.S.-Mexico border," 2013.

[6] Peter Hillyard, Neal Patwari, Samira Daruki, and Suresh Venkatasubramanian, "You're crossing the line: localizing border crossings using wireless rf links," *IEEE Signal Processing and SP Education Workshop (SPW-2015)*, 2015.

[7] Mohamed H. Hashim and Stavros Stavrou, "Measurements and modelling of wind influence on radiowave propagation through vegetation," *IEEE Trans. Wirel. Commun.*, 2006.

[8] Kenneth Bannister, Gianni Giorgetti, and Sandeep KS Gupta, "Wireless sensor networking for hot applications: Effects of temperature on signal strength, data collection and localization," in *Proceedings of the 5th Workshop on Embedded Networked Sensors (HotEmNets 08)*. Citeseer, 2008.

[9] John Thelen, Dann Goense, and Koen Langendoen, "Radio wave propagation in potato fields," in *1st Workshop on Wireless Network Measurements*, 2005, vol. 2, pp. 331–338.

[10] Peter Hillyard, "Border crossing data," 2016.

[11] Ossi Kaltiokallio, Maurizio Bocca, and Neal Patwari, "A fade level-based spatial model for radio tomographic imaging," *IEEE Trans. Mob. Comput.*, vol. 13, no. 6, pp. 1159–1172, 2014.

[12] Yi Zheng and Aidong Men, "Through-wall tracking with radio tomography networks using foreground detection," *IEEE Wirel. Commun. Netw. Conf. WCNC*, pp. 3278–3283, 2012.

[13] Joey Wilson and Neal Patwari, "A Fade-level skew-laplace signal strength model for device-free localization with wireless networks," *IEEE Trans. Mob. Comput.*, vol. 11, no. 6, pp. 947–958, 2012.

[14] H. Hashemi, "A study of temporal and spatial variations of the indoor radio propagation channel," in *PIMRC-94*, Sep 1994, vol. 1, pp. 127–134.

[15] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] Leonard E Baum, John Alonzo Eagon, et al., "An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology," *Bull. Amer. Math. Soc*, vol. 73, no. 3, pp. 360–363, 1967.

[17] Mathew G Pelletier, Sundar Karthikeyan, Timothy R Green, Robert C Schwartz, John D Wanjura, and Greg A Holt, "Soil moisture sensing via swept frequency based microwave sensors," *Sensors*, vol. 12, no. 1, pp. 753–767, 2012.

[18] Moustafa Seifeldin, Ahmed Saeed, Ahmed E Kosba, Amr El-Keyi, and Moustafa Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *IEEE Trans. Mobile Computing*, vol. 12, no. 7, pp. 1321–1334, 2013.

[19] Derek Molloy, "Beaglebone: Video capture and image processing on embedded linux using opencv," 2015.

[20] Santosh Kumar, Ten H. Lai, and Anish Arora, "Barrier coverage with wireless sensors," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2005, MobiCom '05, pp. 284–298, ACM.

[21] Tao Yang, Dejun Mu, Wei Hu, and HuiXiang Zhang, "Energy-efficient border intrusion detection using wireless sensors network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, 2014.

[22] Zhi Sun, Pu Wang, Mehmet C. Vuran, Mznah A. Al-Rodhaan, Abdullah M. Al-Dhelaan, and Ian F. Akyildiz, "BorderSense: Border patrol through advanced wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 3, pp. 468 – 477, 2011.

[23] Xiaolei Li, Qizhen Sun, Jianghai Wo, Manliang Zhang, and Deming Liu, "Hybrid tdm/wdm-based fiber-optic sensor network for perimeter intrusion detection," *Lightwave Technology, Journal of*, vol. 30, no. 8, pp. 1113–1120, April 2012.

[24] D. Tahmoush and J. Silvious, "Remote detection of humans and animals," in *Applied Imagery Pattern Recognition Workshop (AIPRW), 2009 IEEE*, Oct 2009, pp. 1–8.

[25] R.O. Lane and S.D. Hayward, "Detecting personnel in wooded areas using MIMO radar," in *Radar Systems, 2007 IET International Conference on*, Oct 2007, pp. 1–5.

[26] Muzaffar Chishti and C Bergeron, "Virtual border fence given mixed assessment in first test," *Washington, DC: Migration Policy Institute*, 2008.

[27] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, Dec. 2004.

[28] Borys Krawczeniuk, "Carney focuses on illegal immigration," 2008.

PART III

RADIO FREQUENCY SENSING

CHAPTER 5

RUBREATHING: NON-CONTACT REAL-TIME RESPIRATORY RATE MONITORING SYSTEM

5.1 Abstract

The respiration rate of a person provides critical information about their well-being. Conventionally, contact sensing is used for breathing monitoring; however, it is expensive, uncomfortable, and immobile. In-home non-contact breathing monitoring is now possible via Doppler radar and motion capture video sensors, yet these technologies are limited in mobility, among other limitations. When monitoring a patient who is free to move around his or her home, it is difficult to scale current non-contact sensors to cover the large area. Our RUBreathing sensor system uses RF received signal strength (RSS) in a network to estimate breathing rate in real-time with high accuracy over a wide area. In this demonstration, we show the sensor continuously estimating a patient's respiration rate from non-contact RSS measurements between wireless devices. [1]

5.2 Introduction

According to the National Academy of Sciences, an estimated 50-70 million adults in the U.S. suffer chronically from a sleep related issue or disorder. There are over 50 sleep disorders classified by the American Academy of Sleep Medicine with the most prevalent disorders being insomnia, obstructive sleep apnea, restless legs syndrome, and narcolepsy. Individuals who suffer from insufficient sleep are also more likely suffer from additional chronic diseases like depression, diabetes, hypertension, obesity and are at a

[1] I designed and implemented a real-time breathing monitoring system with the CC2530 nodes that were able to explore the limitations, which then inspired me to create the sub-dB system. This section of the chapter appears in IPSN '15: Proceedings of the 14th International Conference on Information Processing in Sensor Networks, "RUBreathing: Non-Contact Real Time Respiratory Rate Monitoring System", Anh Luong, Spencer Madsen, Michael Empey, and Neal Patwari.

higher risk for cancer [1]. A survey conducted in 2008 from the Behavioral Risk Factor Surveillance System provides relevant data on the negative influence insufficient sleep can cause. According to the data , 35.3% reported receiving less than 7 hours of sleep during a typical 24 hour period, 37.9% reported unintentionally falling asleep during the day at least once in the preceding month, and 4.7% reported nodding off or falling asleep while driving at least once in the preceding month. This data, especially the 4.7%, demonstrates the serious consequences from a lack of sleep. In fact, the National Department of Transportation estimates that drowsy driving is responsible for more than 1,500 fatalities and 40,000 nonfatal injuries in the United States each year [2]. Additionally, there are other sleep related issues that would benefit from in-home sleep monitoring. In 2011, approximately 17,000 deaths were caused by drug overdoses of prescription opioid pain relievers. In fact, prescription drug overdose related deaths have increased by 500% since 1990, and are now the leading cause of unintentional injury deaths in the U.S. [3]. Many patients are safely prescribed opioid pain relievers to effectively treat short-term pain. However, opioids cause respiration depression, a decrease in respiration rate and a shallowing of the volume of air inhaled, thereby decreasing the intake of oxygen and causing moments of apnea. If these moments occur frequently or last too long, the oxygen levels will become dangerously low, and serious injury or death can occur. We propose a robust breathing sensor capable of sending an alarm if the breathing level of the patient falls below a certain safety level to be used to provide caretakers or medical personnel the opportunity to intervene and prevent fatality or irreversible damage to the patient. It is critical that the sensor be able to monitor breathing anywhere in the home, as people who are in pain and on pain medication often sleep places other than their bed; for example, it may be more comfortable for a post-surgical patient to sleep in a reclining chair than in a bed.

5.3 System Design

RUBreathing, in this demonstration, utilizes two Texas Instruments CC2530 evaluation modules, each of which contains a system-on-chip, Intel 8051 and Zigbee 2.4GHz radio. This platform is chosen because little computation or memory is required for the RUBreathing sensor, and this platform is low cost and low power.

In order to efficiently sample multiple frequency channels, we use multi-SPIN [4], a time division multiple access (TDMA) protocol in which each node takes a turn broadcasting a packet while the other measures the RSS. Each packet contains the node's latest measured RSS values. After each round of transmission, nodes switch synchronously to the next frequency channel and continue the cycle. A connected sink node overhears all the traffic and records and processes the data.

The RSS is received as a raw stream of data as shown in Figure 5.1. Pre-filtering is a low-pass filter that removes unwanted frequency components above 0.4 Hz. Following the filter, the mean is subtracted from the signal to remove the unwanted DC component. In parallel, the motion detector identifies any period containing movement. Mean removal and breathing rate estimation is performed only during periods with no motion. The average power spectral density is computed, and its peak becomes the breathing rate estimate [5].

In a simple real-time implementation, the RSS measurement stream is stored in the buffer *rss* which is *buffL* in size. Each new RSS value is pushed onto the *rss* buffer and the oldest value is popped off. At each new sample, the breathing rate is re-estimated.

5.4 Preliminary Results

In a test, a person lies down on a cot which is 15 cm above the ground. The two sensors are also placed 15 cm off the ground. They are 1.4 m away from the subject's shoulder. The experiments are run in a cluttered room of size 6.1 m by 6.2 m. The subject was alone in this room for the duration of the experiment. In order to create ground truth, the subject is breathing in and out with a metronome at a known fixed rate of 15 breaths per minute (bpm).

People often toss and turn while sleeping. Changes in sleeping orientation can degrade the accuracy of other breathing monitoring methods. In this experiment, we test three different positions including lying face up, lying face down, and lying on the right side. Due to reciprocity, we expect lying on the left side or right side would yield similar results. Figure 5.2 shows a 60 seconds window of the rate estimates. From the figure, the maximum error is 0.4 bpm. The results show little effect of lying orientation on the accuracy of breathing rate estimation.

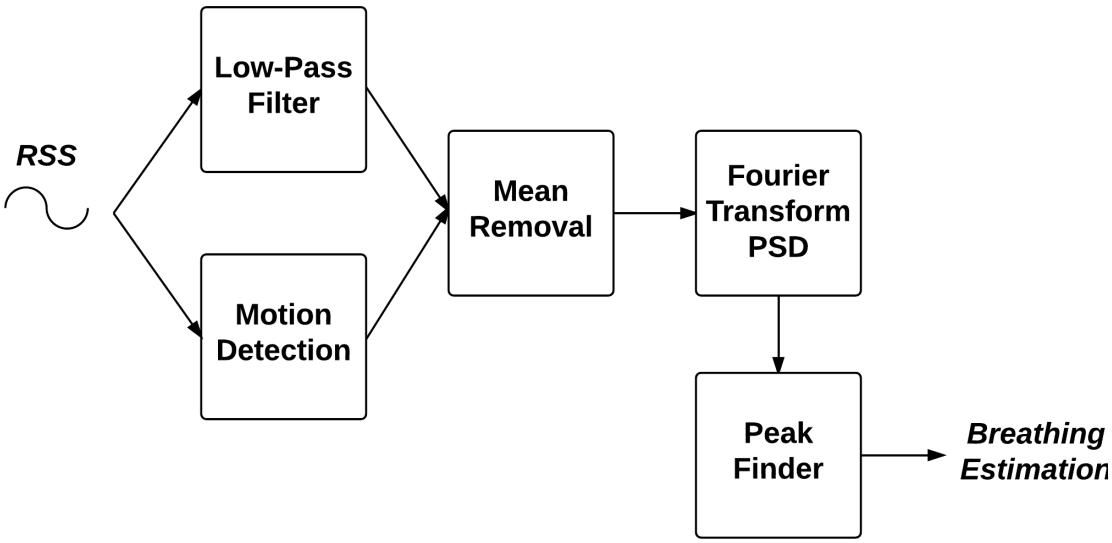


Figure 5.1. Algorithm overview: The raw RSS data stream is used for motion detection and is low-pass filtered. When no motion is present, the mean is subtracted, and the peak of the averaged power spectral density is used as the breathing rate estimate.

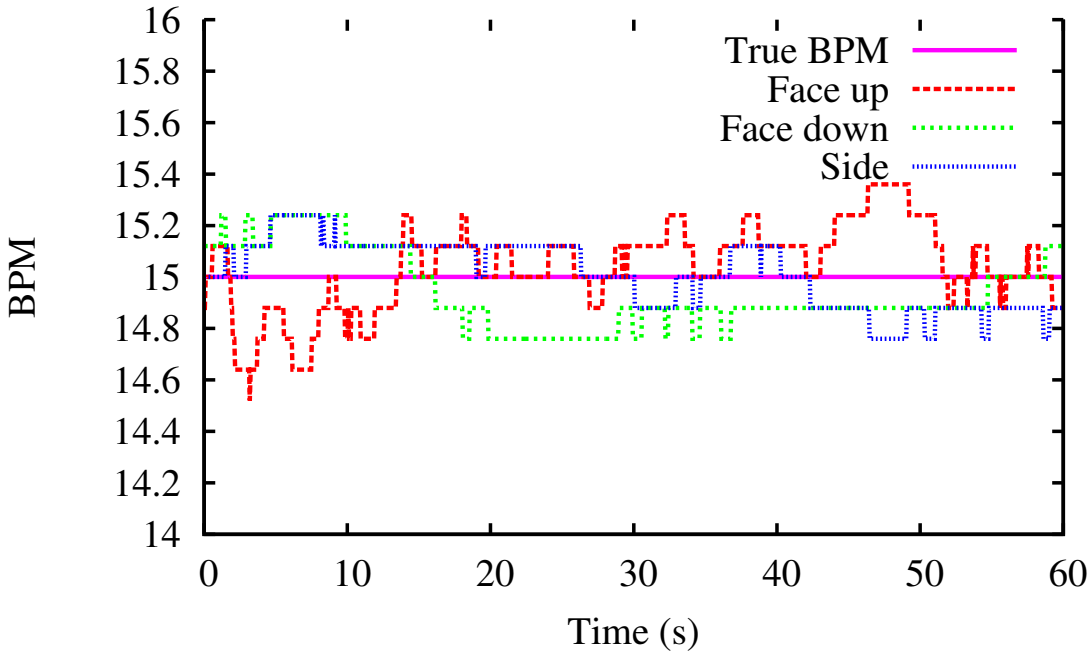


Figure 5.2. Differences between laying position. In this setup, the subject lies face up, face down, and on his side. The channels are affected the same way and breathing rate estimation remains accurate.

5.5 References

[1] Bruce M Altevogt, Harvey R Colten, et al., *Sleep Disorders and Sleep Deprivation:: An Unmet Public Health Problem*, National Academies Press, 2006.

[2] National Center on Sleep Disorders Research US Department of Transportation, National Highway Traffic Safety Administration, National Heart Lung, and Blood Institute, "Drowsy driving and automobile crashes kernel description," 1996.

[3] Centers for Disease Control, Prevention (CDC, et al., "CDC grand rounds: prescription drug overdoses-a US epidemic.," *MMWR. Morbidity and Mortality Weekly Report*, vol. 61, no. 1, pp. 10, 2012.

[4] Maurizio Bocca, Ossi Kaltiokallio, and Neal Patwari, "Radio tomographic imaging for ambient assisted living," in *Evaluating AAL Systems Through Competitive Benchmarking*, Stefano Chessa and Stefan Knauth, Eds., vol. 362 of *Communications in Computer and Information Science*, pp. 108–130. Springer Berlin Heidelberg, 2013.

[5] Ossi Johannes Kaltiokallio, Hüseyin Yigitler, Riku Jäntti, and Neal Patwari, "Non-invasive respiration rate monitoring using a single COTS TX-RX pair," in *ACM/IEEE IPSN*, 2014, pp. 59–70.

CHAPTER 6

RSS STEP SIZE: 1 DB IS NOT ENOUGH!

6.1 Abstract

A radio transceiver normally provides received signal strength (RSS) quantized with 1 dB or higher step size. Currently, we know of no application which has demonstrated a need for sub-dB RSS estimates. In this paper, we demonstrate the need for, and benefits of, greater resolution in RSS for breathing rate monitoring and gesture recognition. Measuring RSS requires orders of magnitude less bandwidth than measuring OFDM channel state information (CSI) or frequency modulated carrier wave (FMCW) channel delay. We have designed a prototype with an off-the-shelf low-power transceiver and a processor to achieve an RSS estimate with a median error of 0.013 dB. We experimentally verify its performance in non-contact breathing monitoring and gesture recognition. We demonstrate that simply decreasing the step size of RSS lower than 1 dB can enable significant benefits, enabling extremely low bandwidth RF sensing systems. Results indicate that RFIC designers could enable significant gains for RF sensing applications with four more bits of RSS quantization. [1]

6.2 Introduction

Measurements of the received signal from the links in a static deployed wireless network can be used to monitor people in the area of deployment. Measurements of received signal strength (RSS) have been shown to enable "sensorless sensing" [1], device-free localization [2], [3], activity recognition [4], fall detection [5], border monitoring [6], and breathing monitoring [7]–[9]. The narrowband transceivers used in these systems enable very low cost RF sensor devices, however, RSS contains no information about signal

[1] In this work, I built a prototype to collect sub-dB measurement and evaluated the system with breathing monitoring application. This section of the chapter appears in Proceedings of the 3rd Workshop on Hot Topics in Wireless (HotWireless) 2016, "RSS step size: 1 dB is not enough!", Anh Luong, Alameyehu Solomon Abrar, Thomas Schmid, and Neal Patwari.

phase, and measures the channel at the single frequency channel used to send the packet, and is quantized with a step size of 1 dB or higher.

The RSS on one frequency channel may not show evidence of the event being monitored, e.g., a line crossing or breathing, thus systems have measured multiple channels for reliability. For example, four to sixteen channels are measured in [3], [6]–[8], improving performance but increasing bandwidth usage. WiFi channel state information (CSI) measures tens of channels but requires 20-40 MHz RF bandwidth [10], [11]. Note that CSI includes phase per subchannel but random phase changes between packets [11] make it unusable. Ultra wideband impulse response (UWB-IR) or frequency modulated carrier wave (FMCW) transceivers can measure multidimensional complex-valued channel response for RF sensing; requiring 2.5 GHz for UWB-IR in [12] or 1.8 GHz for FMCW in [13]. In comparison, the 2 MHz RF bandwidth of an IEEE 802.15.4-compliant transceiver, or the few 100 kHz RF bandwidth of a TI CC1200 sub-GHz transceiver, is one to four orders of magnitude more efficient in their use of the spectrum per channel measurement. Since RF sensing systems must operate in and occupy the same spectrum as RF communications systems in order to measure the channel, there is a strong need to have systems operate efficiently.

In this paper, we propose a fundamentally different approach. We propose that having a smaller step size in the RSS measurement allows highly reliable RF sensing while using only one narrowband channel. Few low-power transceivers currently provide access to RSS with resolution better than 1 dB as wireless communications systems do not require it. We explore a transceiver that does, the TI CC1200, a sub-GHz transceiver. We demonstrate the capabilities of single-channel RSS measurements as a function of resolution. We argue that the benefits of a few extra bits of RSS are significant in RF sensing applications, particularly as commercial use and thus bandwidth usage of RF sensing increases.

We emphasize in this paper the benefits for non-contact RF-based breathing monitoring. In an otherwise stationary environment, a person's inhalation and exhalation causes a periodic change to the radio channel that can be observed in the RSS signal. For home health care and "quantified self" applications, it would be very useful to be able to track vitals signs regardless of where a person travels within a coverage area. However, the amplitude of the breathing-induced RSS signal varies unpredictably by frequency chan-

nel, and is often smaller than 1 dB [7]. In [8], a successful approach was demonstrated in which 1) the channel was sampled at a very high rate and then lowpass filtered to increase the effective resolution of RSS; and 2) RSS measurements were performed on sixteen frequency channels in order to increase the likelihood that at least one channel would observe a strong breathing-induced signal. The combination of these two effected reliable breathing monitoring with only two devices, compared to the 30 used in [7], but both increase the utilization of the channel. In this paper we show that neither technique is required if better resolution RSS measurements are available. Systems using laboratory-grade instrumentation have been used to measure breathing rate from high resolution RSS at 60 GHz [9], but this paper is the first we are aware of to use an inexpensive wireless transceiver IC, and to explicitly show the benefit of sub-decibel resolution.

We also address the impact of sub-dB RSS resolution on gesture recognition. When the distance between the person and the transmitter or receiver is large, the magnitude of the disruption in the RSS is small. If RSS is quantized to 1 dB, the disruption either may not be measured at all, or the spectral features calculated from the quantized RSS may be too noisy. One solution for power-based gesture recognition is to use analog circuitry [14], avoiding the quantization problem altogether. We suggest an alternate approach, and show the ability to see gesture patterns in RSS when the person is several feet from either the transmitter or receiver.

6.3 Sub-dB RSS Measurement

To measure RSS with step size less than 1 dB, we build a prototype system from off-the-shelf components, shown in Figure 6.1, which we refer to as the *sub-dB RSS measurement system*. Our prototype uses a TI CC1200EMK-420-470 evaluation module board connected to a Beaglebone Black (BBB). The CC1200 is a transceiver able operate at 169 MHz, 434 MHz, or 900 MHz. Our evaluation board was populated with a 420 – 470 MHz matching network. We use a TI EM Adapter BoosterPack to bridge the CC1200 board to the BBB.

The CC1200 datasheet states that 12 bits of RSS (equivalently, 1/16 dB step size) is available, but this is misleading. Our empirical study shows that the least significant 4 bits do not change with sub-dB changes in the received power. Instead, the four LSBs

only change when there is a change in the AGC gain stage, that is, the gain stage correction factor may have 12 bits resolution. However, the RSS on the CC1200 effectively has 1 dB quantization steps.

As the CC1200's native RSS would not provide the desired resolution, our sub-dB system relies on another feature of the radio, the IQ sample feature. The CC1200 exports three registers for the magnitude and two registers for the angle of each sample immediately after the CORDIC algorithm. We denote the nth sample as s_n. The sampling period is determined by the channel bandwidth setting. In our case, a new magnitude $|s_n|$ comes through the 7 MHz SPI bus every 22.2 μs, a sample rate of 45.044 kHz. The data is meaningless if the sample is read out while the buffer is being written with another sample. Thus, we check the CC1200 `MAGN_VALID` signal, which rises in order to indicate the availability of the new measurement.

Due to the preemptive architecture of the Linux OS, we cannot use the BBB's main processor to capture the IQ samples from the radio at a precise regular interval. However, the BBB comes with two real-time co-processors, the programmable real-time processing unit (PRU) sub-system. Our application configures the radio, starts magnitude data collection on the PRU, and writes it into shared memory. The main processor calculates the sum of the squared magnitude $P = \frac{1}{N}\sum_{n=1}^{N} |s_n|^2$. The RSS (in dBm) is then $c + 10\log_{10} P$. The value of constant c is calculated via a calibration experiment in which P is computed while a known signal power is input by cable to the receiver. With $N = 100$ and the time required for configuration and computation, our system has an RSS sample rate of 348 Hz.

6.3.1 RSS Evaluation

We validate sub-dB by analyzing its performance with respect to an input signal with known magnitude function. We generated a signal with a National Instruments vector signal generator in which a 434 MHz carrier wave is modulated to have amplitude that varies as a triangular wave with a period of 1 second and a modulation index of 0.9. This signal, shown in Figure 6.2, is input to the CC1200 receiver directly via cable connection so that we know exactly what signal is received.

We then plot the RSS calculated by sub-dB in Figure 6.2, which shows that the two almost perfectly overlap. If the RSS is quantized to 1 dB, there would be larger differences,

as shown in Figure 6.3. Errors from sub-dB are less than 0.06 dB except for when the received power is lowest, perhaps because the SINR is lowest. The CDF of error for sub-dB shows a median error of 0.013 dB for our system, vs. 0.25 dB for when RSS has a 1 dB step size, a 19x difference.

6.4 Breathing Monitoring

In this section, we evaluate the performance of a breathing rate estimator using measurements from the sub-dB prototype system.

6.4.1 Experiments

We conducted experiments with three different subjects and three different positions. The subject lies on a cot 15 cm above the ground. The antennas are also 15 cm above the ground and 60 cm away from either 1) each shoulder of the subject, or 2) from the head and feet of the subject. All of the above described experiments were done in a office of size 8.4 m $\times$ 6.7 m, cluttered with desks, computers, monitors, and lab equipment. For ground truth, the subject wears an Ambu RIPmate inductance belt around his abdomen. We have included experiments with both controlled and uncontrolled breathing as well. In controlled breathing, the subject is required to breath in and out with a metronome at a fixed rate. Since breathing tends to be heavier when the subject is conscious about his breathing, we include multiple uncontrolled breathing experiments to avoid that bias.

Figure 6.4 shows a 30 seconds window of the RSS, showing periodic variations very similar to the RIP belt, with peak-to-peak difference of 0.1 dB. With RSS quantized to 1 dB, there is little chance that the changes would be observed. In the power-spectral density (PSD) of the sub-dB RSS, however, we can distinctively observe a peak at 0.244 Hz (14.64 bpm). This is only 0.12 bpm different from the peak of the PSD of the RIP belt voltage signal.

6.4.2 Evaluation of Breathing Monitoring

During real-time operation, an application using RSS could operate on the 348 Hz stream of floating-point RSS data. However, for evaluation, it is critical to know how well the system would work with more efficient settings of sample rate and quantization step size. For example, if an 7.25 Hz RSS sample rate is sufficient, we could save energy and

reduce channel utilization by using the transceivers only 1/48 of the time. A key question to be answered is the required RSS step size, or equivalently, the number of bits of RSS. We study both quantitatively via the following procedure.

Each floating-point RSS value and its timestamp from the sub-dB system is saved to file for post-processing. The post-processing algorithm operates as shown in Figure 6.5. We downsample by P to evaluate the performance had we sampled less often, and quantize to Q bits to simulate the performance had the RSS been quantized to a particular step size. Here, $Q = 8$ is defined as a step size of 1 dB, as many RFICs use $Q = 8$ to provide 1 dB step size. Each additional bit cuts the step size by half. Filtering is used for two reasons: 1) a DC-removal filter removes the mean so that we look only at the changes in RSS; and 2) as in [8], a lowpass filter provides a means to approximate a higher resolution quantizer when oversampling — we include it for comparison purposes. We then downsample by $M = 144/P$ so that the rate of 2.4 Hz is used in the breathing rate algorithm, regardless of P. This ensures a constant computational complexity for the discrete Fourier transform (DFT). We use a window of the most recent 30 seconds of data as input to the DFT. The breathing rate estimate is the frequency with the maximum amplitude of the DFT (equivalently, the PSD).

Many medical monitors count the cycles in the signal to estimate breathing rate. We implemented a version we call *count breaths* (CB), which counts zero crossings over the same 30 s window. However, by counting only whole numbers of breaths, CB quantizes the breathing rate estimate, and thus introduces quantization error of its own.

6.4.3 Experimental Results

We provide a comparison of RSS-based and RIP belt-based breathing rate estimates from both metrics in Figure 6.6, as a function of the downsampling rate P. Having $P \leq 12$ (for a raw RSS sampling rate of 21.75 Hz) results in relatively constant performance, with estimates within about 0.1 bpm of the RIP belt, on average. We note average error stays below 1 bpm until the sampling rate falls below 7 Hz. In [8], an MA error of 0.12 bpm was achieved, but using 16 Zigbee channels (across 80 MHz) each sampling RSS at 31.25 Hz. We achieve similar performance transmitting only on one channel (across 12.5 kHz) with 30% fewer RSS samples. Sub-dB quantization allows the transmitter to transmit less often

and use 6000 times less bandwidth.

Sub-dB provides floating point RSS. However, breathing monitoring does not require this precision. We quantized the sub-dB RSS measurement to Q bits, for $8 \leq Q \leq 16$, when $P = 12$. The results, shown in Figure 6.7, show that breathing rate estimation does not gain in accuracy as the number of quantization bits exceeds 12 bits. We note that 12 bits corresponds to $1/16$ dB step size.

6.5 Gesture Recognition

Another application of sub-dB RSS is gesture recognition. When a person makes a gesture with their arms or legs in the presence of a wireless link, they cause a temporal pattern in RSS measurements that depends on the action [4]. If the gesture is very close to a receiver, the received power changes can directly be used for gesture recognition [14]. However, when the person is distant from either transceiver, the state-of-the-art is to use micro-Doppler from OFDM for gesture recognition [15]. In this section, we show that RSS changes are small but observable with sub-dB RSS, despite a person's distance from the transceivers.

In our setup, we place two antennas 6.7 m apart. The person stands 2.4 m from the midpoint of the line between the devices. We test four gestures:

- *Punch*: Move arm forward and return it swiftly
- *Kick*: Move foot forward and return it swiftly
- *Zoom in & zoom out*: Stretch both arms wide open and bring them back together
- *Bowling*: Move right arm back and forward in a smooth motion while keeping torso low and one leg back

Each gesture has its unique signature in the RSS measurements as shown in Figure 6.8. However, the peak-to-peak change is always below about 0.75 dB. Based on that, we can see that an RSS measurement quantized to the nearest integer dBm would not provide much information about the gesture.

To quantitatively evaluate gesture recognition as a function of quantization, we use the following algorithm. We apply a support vector machine (SVM) classifier (purely

for proof-of-concept). We select twenty features, including variance, skewness, eight percentiles evenly spaced from the 5^{th} to 95^{th} percentiles, and power spectrum in three different bands. We train the SVM classifier using a set of 43 trials of each gesture, and test using a set of 110 unlabelled gestures. In Figure 6.9, if we train and test on the two gestures "punch" and "zoom in & zoom out", the accuracy of classification improves from just above 50% with eight bits to 100% at 12 bits. With all four gestures, the performance similarly increases from 31% to around 85% at twelve bits, but the performance does not improve higher than 86%. In summary, as a proof-of-concept, we believe that gestures performed far from either transceiver can be recognized from link RSS measurements if the resolution of the measurements is twelve bits (equivalently, 1/16 dB step size) or higher.

6.6 Conclusion

In this paper, we describe sub-dB, a prototype using a TI CC1200 which exposes floating point RSS measurements to the application. Sub-dB estimates RSS with 0.013 dB median error, compared to 0.25 dB median error from a typical 1 dB step size RSS measurement. We evaluate the performance of RSS-based breathing monitoring and gesture recognition.

For RSS-based breathing monitoring, we achieve similar rate estimation performance to [8] but using 6000 times less bandwidth. Sub-dB enables gestures to be recognized from RSS even when the person is far from either transceiver. We demonstrate that twelve bits of RSS, rather than the typical eight bits, provides enough resolution for both gesture and breathing monitoring. This work provides motivation for RFIC designers to provide a few more bits of RSS resolution, as it requires only a few extra logic gates. Additional registers may be required for these bits, but applications that do not require high resolution RSS can ignore them.

Future work should directly compare the application performance when using sub-dB vs. other wider bandwidth channel measurements. CSI and UWB may be advantageous for the application but pose scalability challenges. This paper has demonstrated that performance of narrowband RF sensing can be improved simply with higher resolution power measurements.

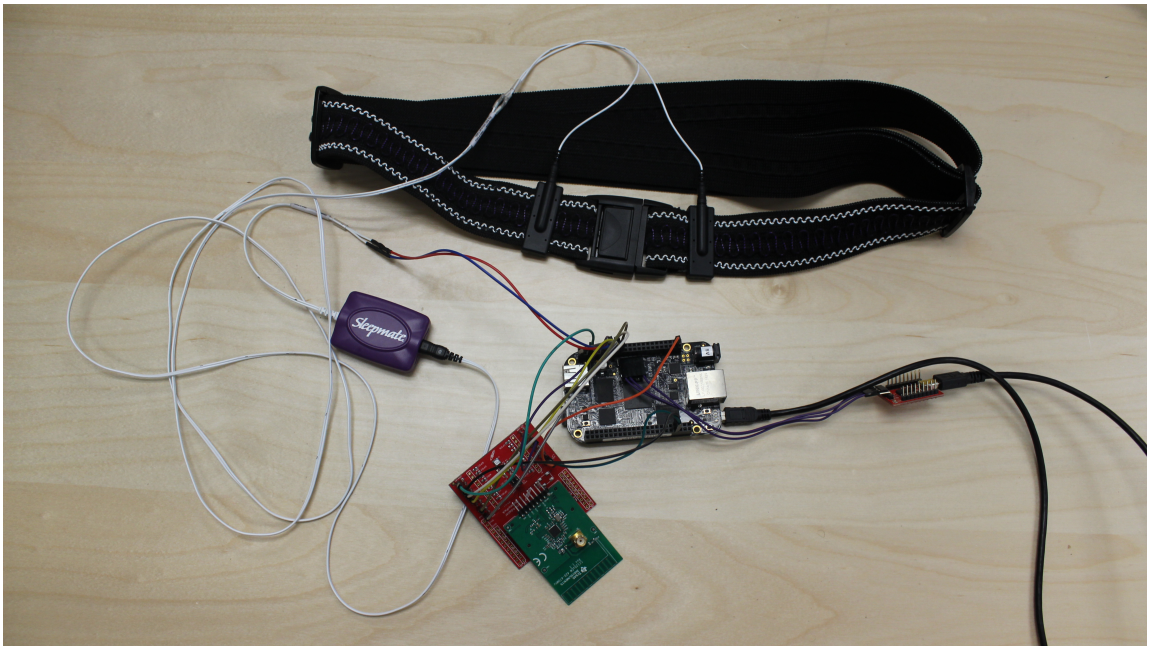


Figure 6.1. Sub-dB prototype: Beaglebone Black and CC1200 eval module. Ambu RIP band provides ground truth breathing rate.

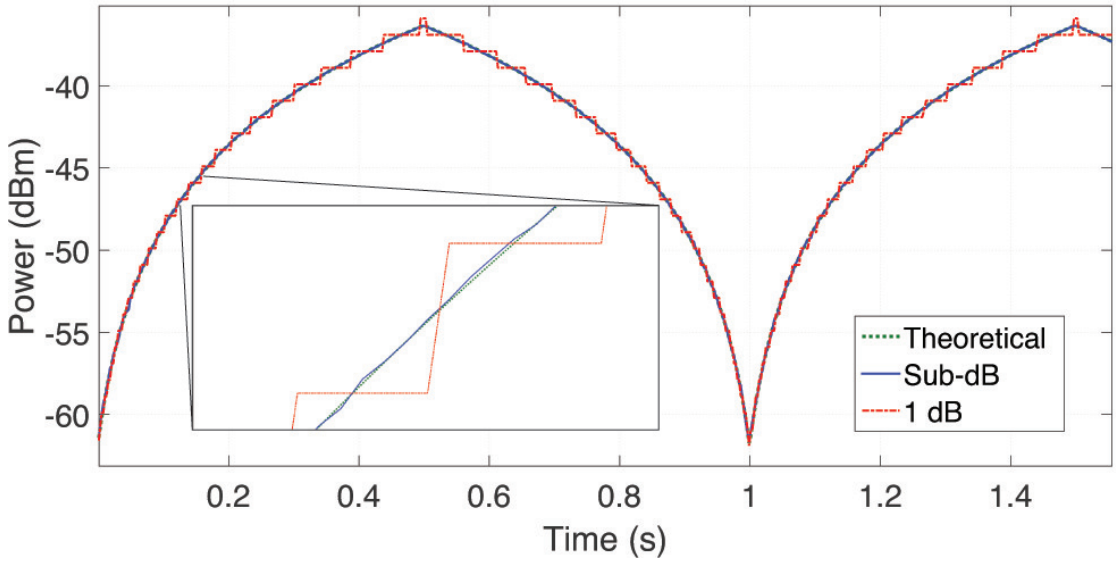
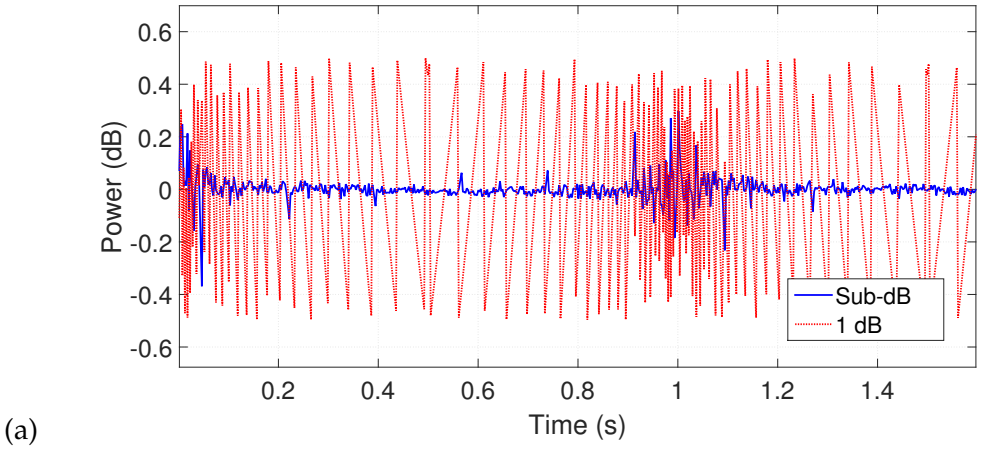
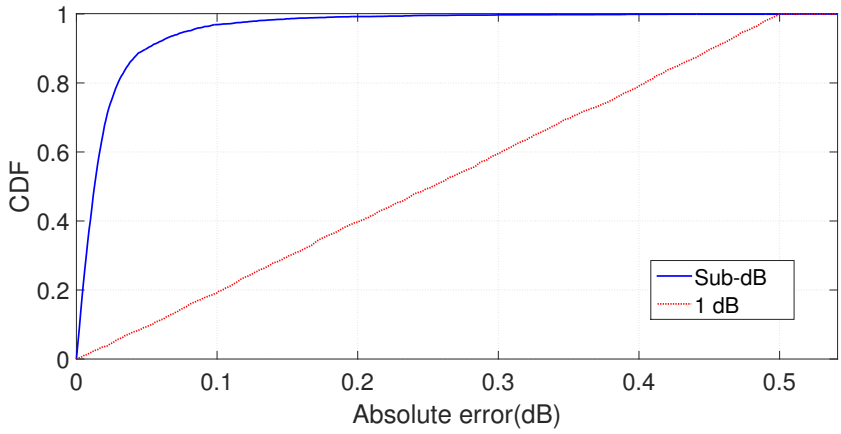


Figure 6.2. Power (dBm) of triangle wave signal input to the CC1200 (· · ·), measured by proposed system (——), and if quantized to 1 dB (- - - -).



(a)



(b)

Figure 6.3. RSS error of sub-dB system (——) vs. RSS quantized to 1 dB (——), (a) over time, and (b) CDF.

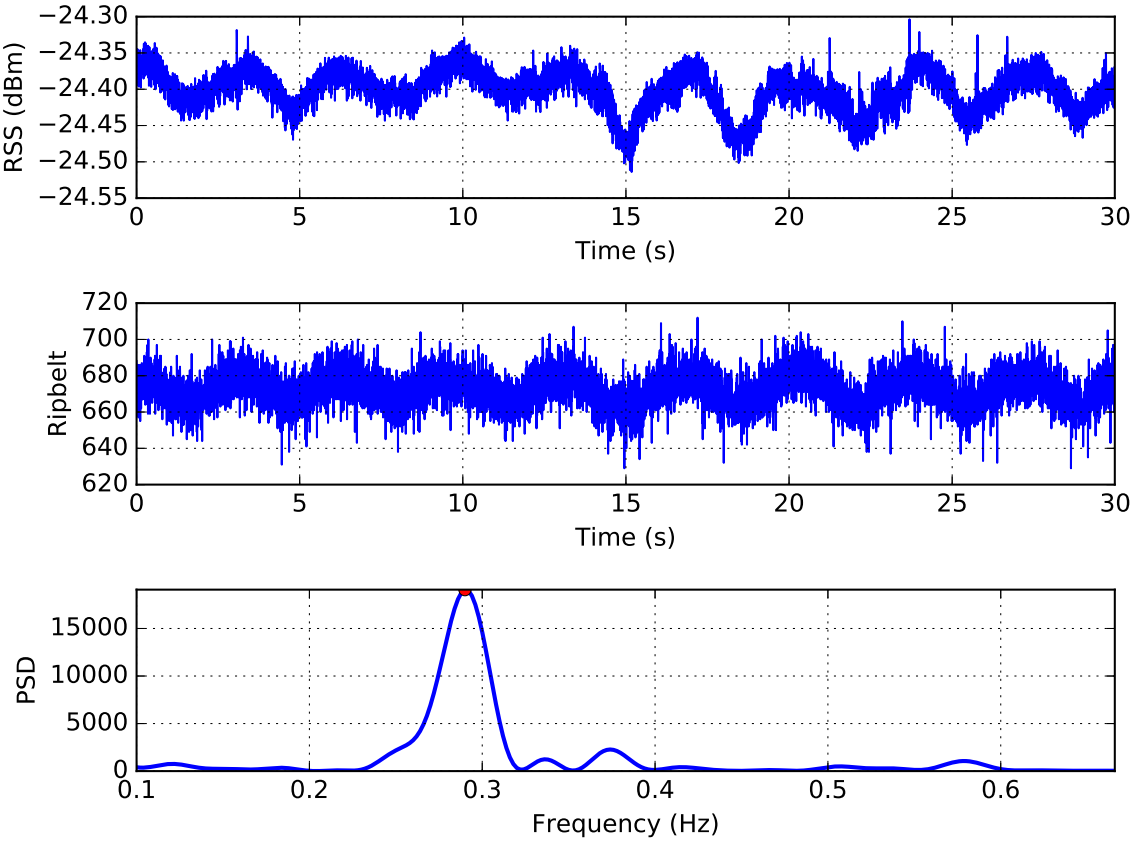


Figure 6.4. (Top) Breathing-induced RSS changes compared to (Middle) RIP belt measurements. (Bottom) PSD of filtered RSS w/ rate estimate (●).

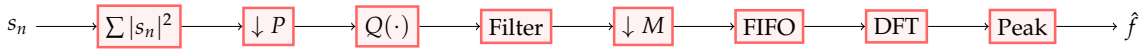


Figure 6.5. Post-processing of RSS data to allow evaluation w/ lower sampling rate & different quantization.

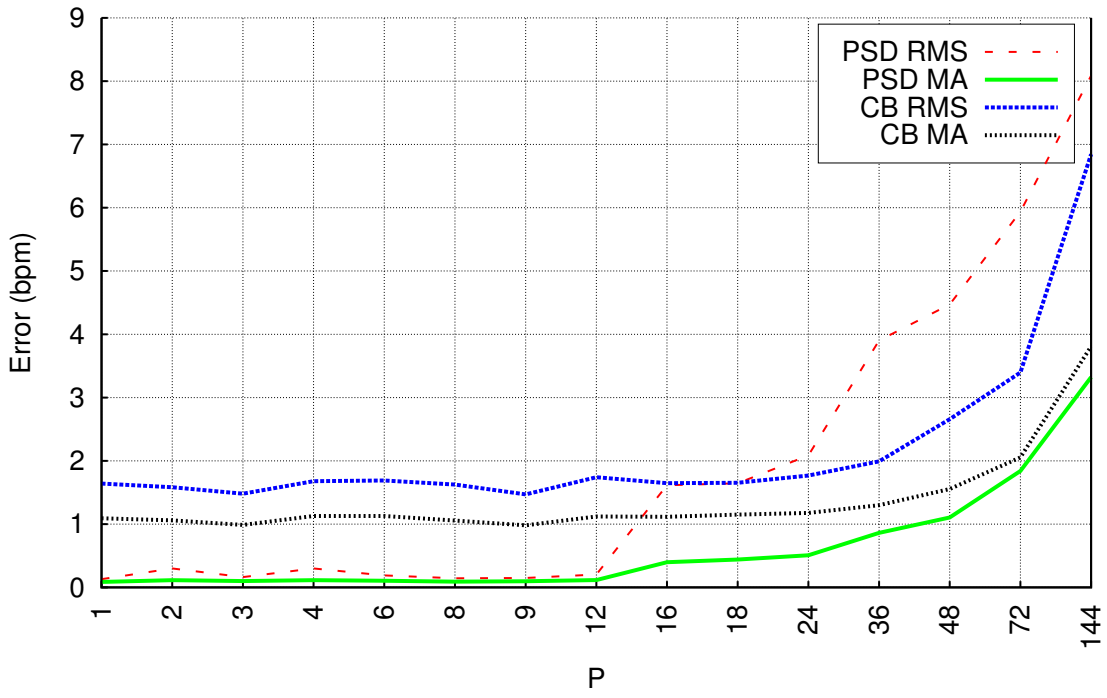


Figure 6.6. Breathing rate RMS and average (MA) error vs. downsampling rate P. The effective RSS sampling rate is $348/P$ Hz.

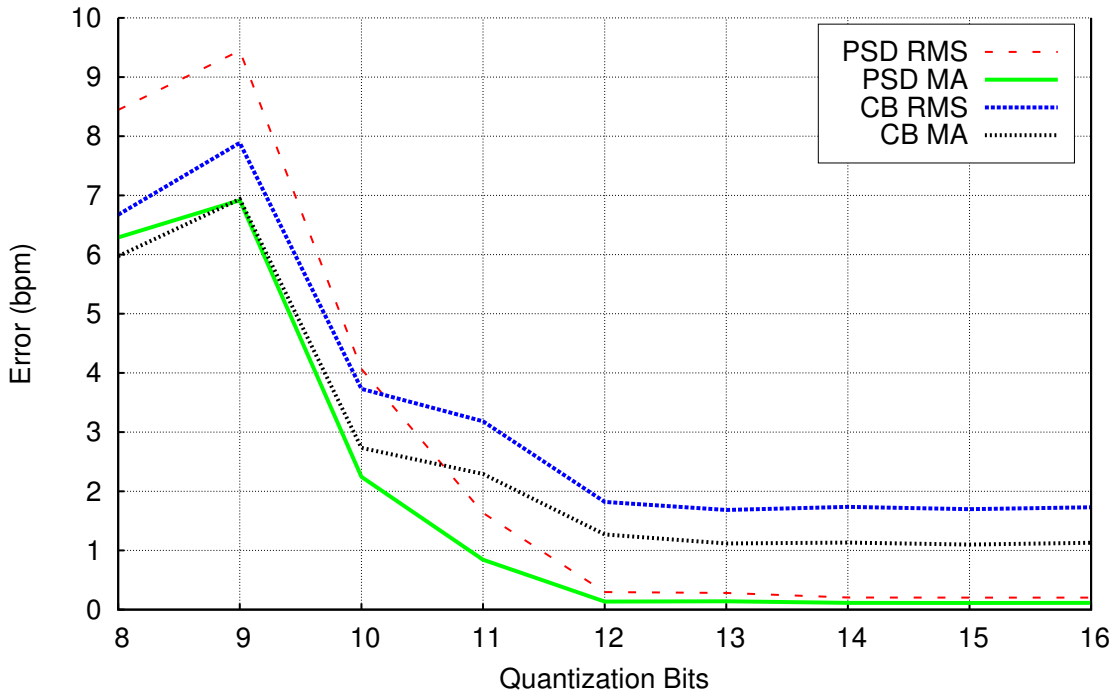


Figure 6.7. Breathing rate RMS and average (MA) error vs. RSS bits ($Q = 8$ means 1 dB step size).

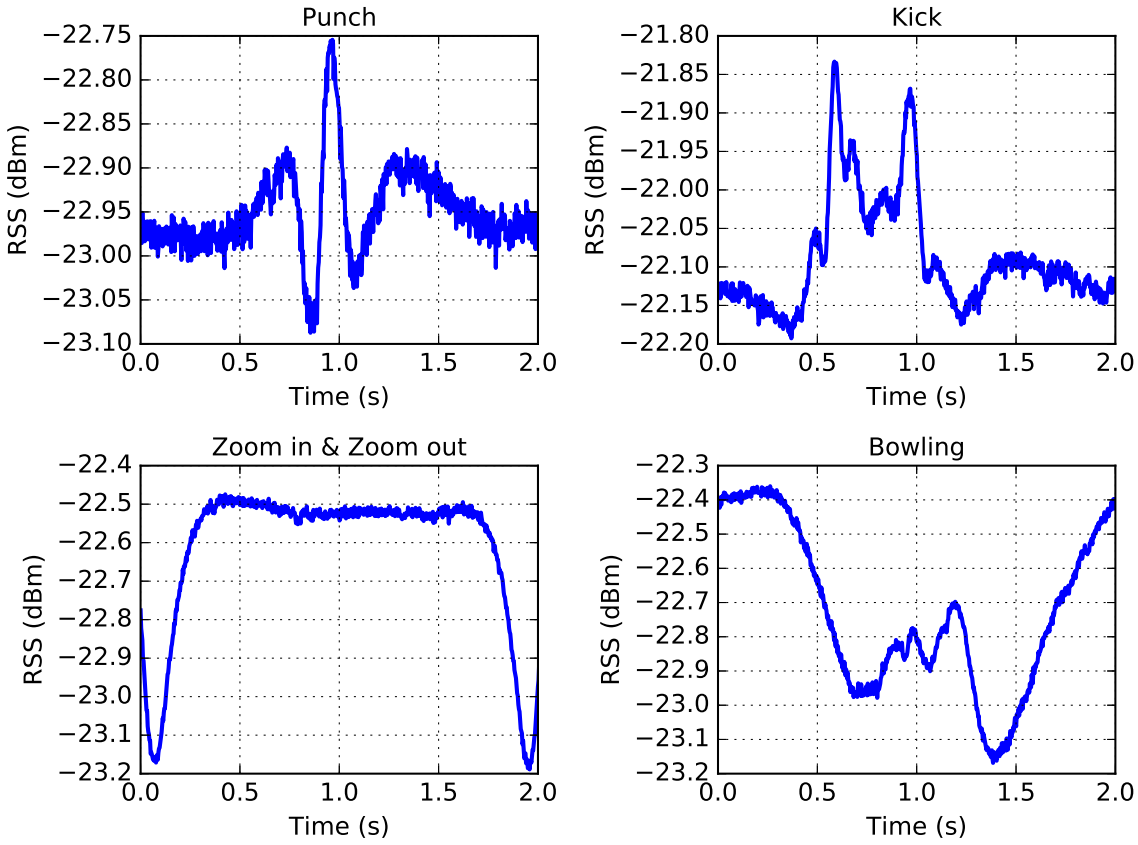


Figure 6.8. RSS signals for four different gestures.

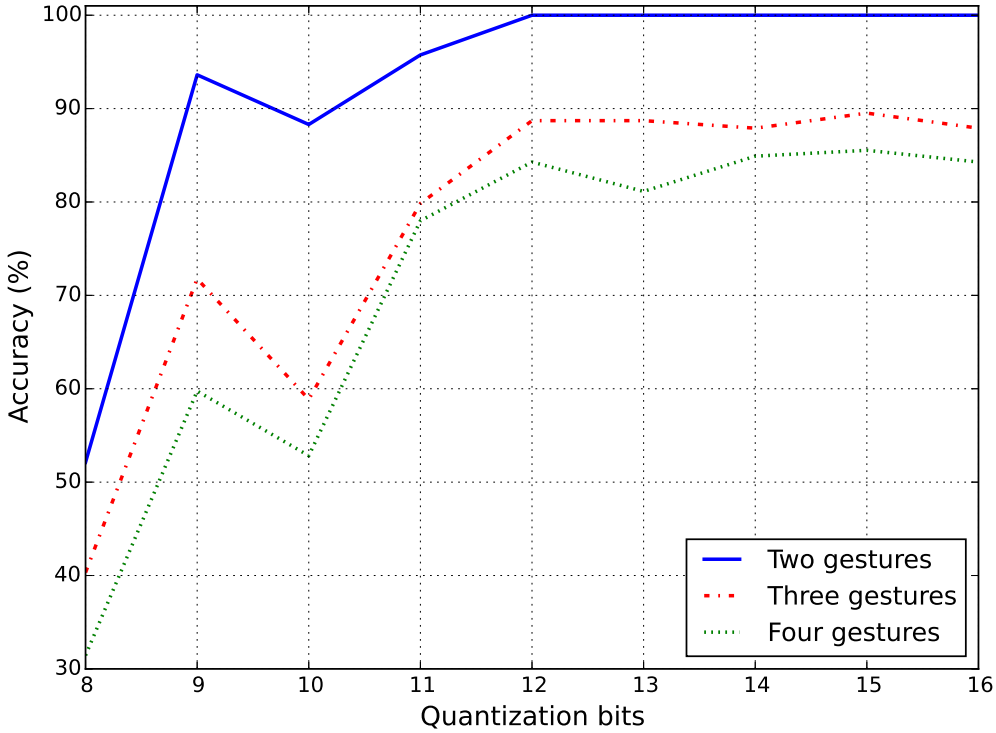


Figure 6.9. Classification accuracy vs. # of bits of RSS and gestures considered.

6.7 References

[1] Kristen Woyach, Daniele Puccinelli, and Martin Haenggi, "Sensorless sensing in wireless networks: Implementation and measurements," in *WiNMee 2006*, Apr. 2006, pp. 1–8.

[2] Chenren Xu, Bernhard Firner, Robert S Moore, Yanyong Zhang, Wade Trappe, Richard Howard, Feixiong Zhang, and Ning An, "SCPL: indoor device-free multi-subject counting and localization using radio signal strength," in *ACM/IEEE IPSN*, Apr. 2013, pp. 79–90.

[3] O. Kaltiokallio, M. Bocca, and N. Patwari, "Follow @grandma: long-term device-free localization for residential monitoring," in *7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2012)*, October 2012.

[4] Markus Scholz, Till Riedel, Mario Hock, and Michael Beigl, "Device-free and device-bound activity recognition using radio signal strength," in *Proceedings of the 4th ACM Augmented Human International Conference*, 2013, pp. 100–107.

[5] Brad Mager, Neal Patwari, and Maurizio Bocca, "Fall detection using RF sensor networks," in *IEEE PIMRC*, London, Sept. 2013.

[6] Peter Hillyard, Anh Luong, and Neal Patwari, "Highly reliable signal strength-based boundary crossing localization in outdoor time-varying environments," in *ACM/IEEE IPSN*, April 2016, pp. 1–12.

[7] Neal Patwari, Lara Brewer, Quinn Tate, Ossi Kaltiokallio, and Maurizio Bocca, "Breathfinding: A wireless network that monitors and locates breathing in a home," *IEEE J. Sel. Topics in Signal Processing*, pp. 30–42, Feb. 2014.

[8] Ossi Johannes Kaltiokallio, Hüseyin Yigitler, Riku Jäntti, and Neal Patwari, "Non-invasive respiration rate monitoring using a single COTS TX-RX pair," in *ACM/IEEE IPSN*, 2014, pp. 59–70.

[9] Zhicheng Yang, Parth H Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra, "Monitoring vital signs using millimeter wave," in *ACM MobiHoc 2016*, July 2016.

[10] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall, "Tool release: gathering 802.11n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[11] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *ACM MobiCom*, 2015, pp. 65–76.

[12] Yakup Kilic, Henk Wymeersch, Arjan Meijerink, Mark J Bentum, and William G Scanlon, "Device-free person detection and ranging in UWB networks," *IEEE J. Sel. Topics in Signal Processing*, vol. 8, no. 1, pp. 43–54, 2014.

[13] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C Miller, "Smart homes that monitor breathing and heart rate," in *ACM Conf. Human Factors in Computing Systems*, 2015, pp. 837–846.

[14] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota, "Bringing gesture recognition to all devices," in *USENIX NSDI*, 2014, pp. 303–316.

[15] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel, "Whole-home gesture recognition using wireless signals," in *ACM MobiCom*, Sept. 2013, pp. 27–38.

PART IV

TIME AND FREQUENCY SYNCHRONIZATION

CHAPTER 7

A STITCH IN TIME AND FREQUENCY SYNCHRONIZATION SAVES BANDWIDTH

7.1 Abstract

Frequency synchronization is a critical requirement for wireless clock synchronization and time-based localization systems. Existing solutions use multiple time offset measurements for estimation of clock skew, which requires significant channel utilization. In particular, the limited bandwidth available for ultra wideband (UWB) sharply limits the scalability of UWB-based systems. We introduce and evaluate a platform architecture, *Stitch*, which allows the frequency and time synchronization of multiple nodes' clocks. The platform accesses complex baseband samples from a low-power, low-cost narrowband radio, estimates the carrier frequency offset, and iteratively drives the difference between two nodes' main local oscillators (LO) to less than 3 parts per billion (ppb). We implement optimized time synchronization and ranging protocols that achieve the same timing accuracy as the state-of-the-art but with 59% less utilization of the UWB channel. [1]

7.2 Introduction

Clock synchronization is a fundamental requirement for efficient operation of a wide variety of wireless networks. Multiple access methods, including code, frequency, and time division multiple access, can all be made more efficient when a large-scale network of devices is synchronized. A wide range of time-based localization systems require

[1] In this joint work, I designed and introduced the Stitch achitecture, which I built in the previously described Chronos platform upon. I also implemented and evaluated the RFS algorithm for frequency synchronization of two independent oscillators. With that, I was part of the designing and implementing of EffToF, a protocol that efficiently estimate time-of-flight measurement for synchronization or ranging. Stitch, Chronos, RFS, and EffToF are all evaluated in this chapter. This chapter is also unpublished, "A Stitch in Time and Frequency Synchronization Saves Bandwidth", Anh Luong, Peter Hillyard, Alameyehu Solomon Abrar, Charissa Che, Thomas Schmid, and Neal Patwari.

time-synchronized infrastructure devices deployed across space: although cables can be used to distribute a shared clock to infrastructure devices, cabling is expensive and often impractical; GPS-based synchronization has limited availability when operating indoors or when it is jammed. Thus, wireless clock synchronization is highly desirable, but it is itself expensive, as it requires the use of the radio channel for synchronization, above and beyond the use of the channel for communication or localization. This paper makes contributions to dramatically increase the efficiency with which synchronization can be performed. By increasing synchronization efficiency, we can increase the availability of the channel for its primary purpose, for example, allowing higher throughput, or increasing the number of devices which can be located.

This paper evaluates our contributions to wireless synchronization in a time-based ranging system, and specifically, one using ultra-wideband impulse response (UWB-IR) signals. Such UWB-IR systems are widely seen as a means to achieve highly-accurate localization [1]; however, UWB-IR transmission occupies GHz of spectrum, and due to the bandwidth allocated to UWB [2], the UWB-IR channel is quite limited in terms of packet rate. For logistics applications involving the need to track millions of objects, bandwidth and channel utilization are critical limiting factors. We provide a system that achieves the same ranging accuracy as the state-of-the-art UWB-IR system with 59% less utilization of the UWB channel.

This paper achieves this improved efficiency with three key novel contributions to the synchronization and time-based localization literature across a broad range of wireless networking applications:

1. We introduce *Stitch*, an architecture that allows clock unification and adaptive distribution across a platform's subsystems. This proposed architecture allows the synchronization of one subsystem to be propagated to the entirety of a platform.

2. We design, implement, and evaluate a new *wireless frequency synchronization system*, Radio Frequency Synchronization (RFS), which uses low-cost narrowband (NB) transceivers to measure the frequency offset and to synchronize the local oscillators (LOs) on two devices to be within 3 ppb of each other. Stitch allows this highly syntonized LO to be shared across transceivers.

3. We implement and evaluate a new protocol, *EffToF*, which combines RFS with minimal use of a UWB radio to achieve *bandwidth-efficient wireless time synchronization and time-of-flight (ToF) ranging*.

We implement the three contributions on a research platform we call the *Chronos* board. Chronos implements the Stitch architecture and provides the ability to adapt the clock distribution network in software. We demonstrate clock unification among multiple subsystems (microprocessors, FPGA, Texas Instruments CC1200, and DecaWave DW1000). We then evaluate the performance of the frequency synchronization and measurement of ToF on a testbed shown in Figure 7.1 and show ranging RMSE of 17.1 cm with standard deviation of 3.15 cm using 59% less of the UWB channel compared to PolyPoint [3]. The Chronos board is a superset of subsystems that could potentially be useful in a range of applications for wireless synchronization research.[2] Any particular application, such as the time-based ranging discussed in this paper, would require only a small subset of the systems contained on the Chronos board.

7.3 Related Work

7.3.1 Message Exchange

There has been a large body of research addressing time synchronization in wireless networks via radio message exchange. The reference broadcast synchronization (RBS) system [4] uses the transmission of a reference message and its reception on the nodes as marker of epoch for time synchronization. However, the protocol does not consider the propagation delay of the message, which can introduce significant error. The timing sync protocol for sensor networks (TPSN) creates a spanning synchronization tree with nodes that perform pairwise synchronization by two-way message exchange [5]. This then allows nodes to mitigate transmitter and receiver bias. TPSN still assumes there clock drift during message exchange is negligible. While an additional message exchange would allow a node to estimate its skew, it would increase the utilization of the channel. Flooding-Time Synchronization Protocol (FTSP), Glossy, and PulseSync address the problem of time synchronization for multihop or large sparse network with constructive

[2]The hardware, firmware, and software are open source and will be available after double-blind review.

interference through synchronous flooding, thus achieving better synchronization [6]–[8]. Fundamentally, clock synchronization still requires large amount of messages to converge on an accurate clock's offset and skew.

7.3.2 XO

Wireless devices rely heavily on oscillators for timekeeping (timestamping, tasks scheduling, etc.). A natural approach to achieve more accurate frequency synchronization across devices is to put a highly stable oscillator on each node. Other stand-alone methods to achieve similar accuracy are: GPS-disciplined oscillators [9]; oven controlled oscillators (OCXO) [10]; and chip scale atomic clock (CSAC) [11]. All are expensive and have high power consumption [12], as shown in Figure 7.2. For example the CSAC in [13] uses 120 mW and costs over 1500 USD (quantity of 1). Such specifications preclude their use in many battery-powered and consumer applications. In this paper, we describe a method for achieving frequency synchronization accuracy on the order of a few parts per billion with low-cost (22 USD quantity of 1) [3] off-the-shelf components and an average power consumption of 7.5 mW.

7.3.3 Reference Reception

A variety of research has addressed frequency synchronization of an oscillator based on a transmitted and received signal. The signal may be ambient or actively transmitted. The ambient electric grid frequency, received by a magnetic field sensor, can train a low-power oscillator [14], but such methods have not been used to synchronize the main local oscillators on different nodes. Similarly, an oscillator's frequency can be actively transmitted as a pilot and received by multiple devices [15]–[17], which is accurate enough to enable distributed beamforming. However, a software defined radio is required for clock synchronization, and spectrum regulations may not allow transmission of the desired LO frequency. Similarly, custom analog RF circuitry could be used to receive the LO frequency signal [18]; however, this adds additional complexity. In comparison, our solution allows transmission at wide range of carriers and is implemented with standard digital integrated circuits (IC).

---

[3] We keep quantiy of 1 for fair comparison with CSAC and OCXO. At quantity of 1000, the cost is 11 USD.

7.3.4 Carrier Frequency Synchronization

Carrier frequency synchronization is already a required part of demodulation on most radio receivers [19]. On the TI CC1200, this feature is called *automatic frequency compensation*. When uncompensated, the received complex symbols rotate in the I-Q plane, and the symbol error rate increases. Receivers estimate and correct the phase and frequency offset of the complex baseband signal in a phase-locked loop (PLL). After demodulation, any estimate of the phase or frequency offset is typically discarded. Most transceiver ICs do not provide an option to alter the frequency of the local oscillator external to the radio or export the estimated frequency or phase offsets. That is, only the radio samples are frequency corrected and only during reception; the rest of the embedded system is unchanged.

Some radios provide a "VCOTUNE" signal to fine tune the external voltage controlled oscillator (VCXO) [20]. Due to the high cost, the on-chip circuitry typically provides a limited tuning range. In case of the DecaWave DW1000, its VCXO tuning range is $\pm$25 ppm with 5 bits resolution, thus, 1.56 ppm step size. SurePoint [21] is able to limit this tuning range and achieve sub-ppm by using appropriate loading register (23 ppm with 30 steps, 0.79 ppm step size). In order to adjust crystal offset on the fly, they repeat the first packet on each channel, thus, more message exchanges. None of these systems can achieve synchronization on the order of parts per billion because of the high step size. Our solution is able to tune the VCTCXO with a 0.02 ppb step size and achieve frequency synchronization of less than 3 parts per billion while staying low budget.

7.4 Stitch

In this section, we describe the Stitch architecture and how we use a low-power narrowband radio, the TI CC1200, to synchronize two devices' LOs. We also present the platform we developed as a research tool for clock synchronization research, and our implementation of EffToF, a more efficient time-of-flight estimation protocol, by combining the timestamping accuracy of an UWB radio with our RFS subsystem.

7.4.1 Architecture

Stitch is an architecture that allows clock unification and RF-based syntonization. The key components of this architecture are:

1. **Adaptive clock synthesis**: a field programmable gate array (FPGA) is used for clock synthesis and distribution,

2. **Controllable reference clock**: a digitally controllable local oscillator that is shared across the entire system, and

3. **Frequency offset forwarding radio**: a transceiver that allows exportation of either frequency offset estimates or the raw complex baseband samples.

An additional benefit of the use of an FPGA for adaptive clock synthesis is that it can also be used as a routing table for signal wiring, which increases the adaptability of the platform for other applications.

When independent clocks for microcontroller and radio are used, a node has disjoint clock domains. This architecture results in uncertainty (quantization error) in timestamped events when an event is generated by one subsystem and timestamped by another [22]. While this is a standard architecture, any clock synchronization of the radio's LO simply would not benefit the whole node.

One solution is single clock source with a wide variety of clock synthesizers to fulfill the requirement for the particular combination of radios and microcontrollers contained in the subsystems. For each combination of ICs and subsystems that could be used, a system designer would need to build new clock hardware, which is an inefficient use of resources. We are motivated to create a research platform that allows clock unification and adaptive distribution of a shared clock across a platform's subsystems, even when we don't know what subsystems may be connected to the main board when it is designed.

For example, a platform might have a 40 MHz LO. Later, we might decide to attach a DecaWave DW1000 cape, but the DW1000 requires a 38.4 MHz LO. With Stitch, we would not have to fabricate a whole new platform to attach this cape. Instead, Stitch allows quick reconfiguration of hardware through reprogramming of the FPGA. Thus, a developer can reroute IOs and synthesize the required operating frequencies for individual subsystems from the main clock with the on-chip PLL. Thus, for the previously

described use case, Stitch allows the DecaWave DW1000 cape to benefit from having an LO that is synchronous with that of the main platform, via a simple FPGA reflash.

We introduce Stitch to provide a playground for a wide range of clock-related research and development, including the unique ability to change and adapt the clock network to tailored applications without fabricating a complete redesign.

7.4.2 Implementation

We implemented Chronos, an ambassador for Stitch, to address adaptive low-cost, low-power platform for sensor network clock-related research, as shown in Figure 7.3. We started by designing the Chronos around a low-power FPGA, Microsemi IGLOO AGL250. We see products such as the Altera SoC, Microsemi SmartFusion, and Cypress PSoC as too large, expensive, and power hungry for the application requirements of the Stitch architecture. The FPGA is the main mechanism for reconfiguration of the inputs and outputs of clocks and other digital signals between various subsystems. Thus, an FPGA does not need to have an on-chip microprocessor or state-of-the-art performance to implement Stitch, and the Microsemi IGLOO meets the requirements. With its 250 MHz 3K logic elements, some additional logic could be implemented if required for an application.

The Microsemi IGLOO FPGA comes with the ability to tune the clock source through its PLL. However, this combination of digital PLL and standard clock source has limited adjustment range. Instead, Chronos uses an external stable 1 ppm 40MHz or 38.4MHz Voltage Controlled Temperature Compensated Crystal Oscillator (VCTCXO) from Abracon, which draws 7.5 mW. We concatenate the two 12 bits channels of the Microchip MCP4922 to create a overlapping 24 bit, effectively 20 bit, digital-to-analog converter (DAC) to control the Abracon VCTCXO. Over the approximately 800 Hz range of the VCTCXO, this 20-bit control provides a step size of 7.63×10^{-4} Hz. For a 40 MHz LO, this corresponds to 0.02 ppb.

For raw processing power, Chronos offers two main options: 1) Beaglebone-compatible headers, or 2) a dedicated real-time microcontroller. The Beaglebone Black is a widely popular open-source platform that can run Linux, thus, it provides access to significant pre-existing resources (source code, toolchains, communication protocol stacks, etc…). The Chronos also includes a 120 MHz Freescale MK22 microcontroller for low-power

real-time applications. The MK22 is a Cortex-M4 with a floating point unit (FPU) and a DSP unit, and for that reason, supports high performance computing. The MK22 offers OpenSDA drag-and-drop programming via an on-board Freescale MK20. Moreover, the on-board joint test action group (JTAG) and serial wire debug (SWD) header, and the Beaglebone-compatible SWD connection can also be used as programming interfaces. The Microsemi IGLOO, however, requires an LC Programmer; hence, FlashPro3/4 has to be used with an additional adapter board.

The Chronos contains a power management integrated circuit (PMIC) to enable the use of a single cell Li-Po battery for a short-term deployment. Alternatively, it can be powered via micro-USB connection, as shown in Figure 7.4.

7.5 Radio Frequency Synchronization

In this section, we describe how we use the TI CC1200 radio to provide wireless syntonization. The CC1200 may be used either as a FSK transceiver or optionally, may transmit and receive a pure tone carrier wave (CW). In either case, we may observe from the received signal the difference between the two devices' carrier frequencies. In this section, we describe using CW transmission and reception for this purpose.

The TI CC1200 does not directly provide an option to alter the frequency of the local oscillator external to the radio. We rely instead on the *IQ sample* feature of the CC1200. This feature allows the CC1200 to export two registers for the angle of each sample after the coordinate rotation digital computer (CORDIC) algorithm. The angle is a 10 bit value, which corresponds to 0.35 degrees resolution. At the nth sample time t_n, the 10-bit phase of the received signal, ϕ_n, can by expressed as,

$$\phi_n = \left(2^{10}\Delta_f t_n + \beta\right) \mod 2^{10}, \tag{7.1}$$

where Δ_f is the carrier frequency offset between the transmitter and receiver, and β is the phase offset. Frequency synchronized clocks can be achieved through driving the differences between LOs on the two devices to zero. In Figure 7.5, we show the minimal structure for wireless syntonization with TI CC1200 and VCTCXO.

The sample rate from the CC1200 is limited by the channel bandwidth setting and the maximum speed of the SPI bus. A new IQ sample can only arrive every $T_s = 22.2\mu$s, a sample rate of 45.044 kHz. The data are invalid if the sample is read out while the

buffer is being written with another sample. Hence, we check the rising edge of CC1200 `MAGN_VALID` signal, which indicates the availability of the new measurement.

We implement RFS with the Beaglebone Black as the processing unit, to demonstrate the feasibility of integrating a Linux-based user application with the Chronos platform. Using the on-board MK22 processor for RFS is also possible, and in fact, more straightforward because it is a real-time processor. The Beaglebone's main processor runs a preemptive Linux OS and thus cannot capture the IQ samples from the CC1200 at a precise regular interval. Therefore, we use one of the two real-time co-processors in the programmable real-time processing unit (PRU) sub-system to collect the timing-dependent samples. The PRU supports a very simplified assembly instruction set, which does not allow integer multiplications required for frequency estimation. In this particular use case, our user application first configures the radio, then has the PRU store the IQ samples into the shared memory for access by the main processor.

7.5.1 Frequency Offset Estimation

From the 10 bit phase measurements, we propose and compare two algorithms to estimate the frequency offset. Both operate on the unwrapped phase integer, which we refer to as p_n, where one cycle corresponds to an integer value 2^{10}, rather than 2π radians. We measure at sampling times $t_n = nT_s$, and we expect that $p_n = 2^{10}\Delta_f nT_s$, where Δ_f is the frequency offset in Hz.

7.5.1.1 Naive Estimator

In the "naive" approach, we use the difference in phase over the entire period of N samples:

$$\Delta_f^n = \frac{p_N - p_1}{2^{10}(N-1)T_s} \tag{7.2}$$

7.5.1.1.1 Reference carrier frequency. We set a National Instruments vector signal generator (NI VSG) to generate a carrier frequency at 434 MHz. Because the NI VSG is itself not perfect, we must measure its actual carrier frequency. With a Keysight 53230A 12-bit frequency counter, we measure the generated frequency to be 434 MHz -62.27 Hz. Not all multiplying factors are possible with a PLL, and with a 40 MHz LO, the closest achievable carrier frequency on the CC1200 is 434 MHz -61 Hz, which occurs

when the internal CC1200 PLL has its multiplier and divisor registers set to the have the factor of 10.849998475. We validate this factor by transmitting a pure carrier wave with the CC1200 after setting the correct factor and the various local oscillator frequencies. With the aforementioned PLL factor, the target local oscillator frequency should be at 40 MHz -0.117 Hz. Note that the CC1200 has various intermediate frequency (IF) settings available, and an IF of 138.88 kHz is used in this setup.

As shown in Figure 7.6, we approach a frequency difference < 0.1 Hz after the sixth iteration of the naive algorithm (w/ $N = 1000$ samples per iteration).

7.5.1.2 Linear Regression Estimator

Since the phase noise in p_n fundamentally limits the performance of the estimation of Δ_f, we implement a low-complexity linear regression method that improves performance in noise, as shown in Figure 7.7. In summary, we use linear regression to estimate the slope of p_n with n, to which Δ_f is proportional. However, because we do not require the y-intercept, and because the x-values are regular sampling times, we can significantly reduce the computational requirements and limit the processing to primarily integer multiplications and additions. Generally, a linear regression slope estimate is given by $\frac{\overline{p_n t_n}}{\overline{t_n^2} - (\overline{t_n})^2}$, where $\overline{x_n}$ denotes the sample average of a sequence x_n for $n = 1, \ldots, N$. Since we care only about the slope, any offset of the sampling time axis does not impact the result. Thus we shift the time axis to be zero-mean, in other words, $t_n = T_s \left(n - \frac{N-1}{2}\right)$. With this, $\overline{t_n} = 0$ and $\overline{t_n^2} = \frac{T_s^2 N(N^2-1)}{12}$. Assuming N is odd, the linear regression frequency estimate simplifies to

$$\Delta_f^{lr} = \frac{12 f_s}{2^{10} N^2 (N^2 - 1)} \sum_{n=-(N-1)/2}^{(N-1)/2} n p_n. \tag{7.3}$$

Calculation of Δ_f^{lr} as given in (7.3) requires N integer multiplies and adds, and one floating point scale factor.

7.5.1.3 IQ Sample Size

Due to the quantized nature of the phase measurements, the naive approach has a frequency resolution of $\frac{45044}{2^{10}(N-1)} = \frac{43.99}{N-1}$ Hz. For example, with $N = 1000$, the carrier frequency offset resolution is 0.044 Hz. However, we observe worse performance even with the linear regression method, as shown in Figure 7.8, due to the noise in the phase

measurement.

7.5.2 LO Frequency Correction

The TI CC1200 generates its carrier frequency by multiplying and dividing its LO. This constant fraction is written by the application of the CC1200's configuration registers based on the desired carrier frequency. The CC1200 transceiver operates in any of the 169 MHz, 434 MHz, and 900 MHz ISM frequency bands. In our Chronos board, the matching network is populated for 434 MHz, which is our carrier frequency. Our PLL multiplier is thus $\frac{434\ MHz}{f_{LO}}$. With this relationship, the LO offset (xo_{offset}) is directly proportional to the carrier frequency offset (Δ_f) by this PLL multiplier. We simply translate this carrier offset into a corresponding LO offset and increment the VCTCXO control voltage to achieve this LO offset.

Prior to syntonization, the Δ_f may be large. Nonlinearities in the DAC and the VCTCXO prevent our algorithm from sending the LO offset to zero in one attempt. Instead, after updating the receiver's LO, the receiver iterates to collect another set of N samples and compute the next LO offset. As seen in Figure 7.8, for $N = 1001$, the remaining LO frequency offset is about 0.1 Hz after two cycles of RFS, which for the 40 MHz LO, corresponds to 2.5 ppb.

The power consumption of the Abracon VCTCXO dominates the power consumption budget of the syntonization procedure, as long as syntonization is performed infrequently. The CC1200 uses 46 mA for transmission of a carrier wave at maximum power (14 dBm), which corresponds to 151 mW. The receiver uses half of this power, 75 mW. The MCP4922 DAC, and processing unit requires 1.5 mW and 2.5 W, respectively. Since each syntonization requires 22 ms, one syntonization each 20 minutes requires 45μW. We justify a 20-minute syntonization period with the 24-hour experiment below. Meanwhile, the Abracon ASVTX-12 requires 7.5 mW.

In addition, most of the components for RFS are relatively inexpensive compared to the other high stability oscillators we compare in Figure 7.2. For quantity one, at the current time, the CC1200, MCP4922, and VCTCXO, cost 6.17, 2.70, and 3.58 USD, respectively. Additional passive components increase the total cost to 22 USD. Moreover, high production manufacturing will significantly decrease the costs, especially for the

passives.

7.5.2.1 24-hour RFS

We first conduct the preliminary experiment to quantify the performance of RFS. In this experiment, the nodes are placed on the desk of a clustered office about 1 m apart. The boards are powered on for some time before each experiment to remove any instability during the warm-up period.

Due to the 38.4 MHz requirement for DW1000, the 38.4 MHz is generated and fed both CC1200 and DW1000. Thus, the CC1200 has a PLL multiplication factor of 11.3020833333. For this experiment, the two LOs of the two Chronos boards remain connected to the frequency counter to keep track of their frequencies.

In Figure 7.9, the two oscillators start substantially fluctuate and skew apart more than 1 Hz (26 ppb) after just 20 minutes of free-running, uncompensated. Next, for the same setup, we perform RFS once each 20 minutes. We achieve an RMSE of 0.136 Hz. The freerunning method has RMSE of 0.773 Hz. Thus, RFS reduces the clock offset to 3.541 ppb, while freerunning is 20.128 ppb, more than 5$\times$ higher, as shown in Figure 7.10.

7.6 Efficient Time-of-flight

In time synchronization, we rely on the accuracy of timestamps at the reception and transmission of radio messages in order to estimate the offset and skew between the two clocks through computations with ToF. Thus, a better ToF estimation would greatly improve the estimation of the offset and skew, accuracy of time synchronization. Even though UWB-IR radios provide highly accurate timestamps, these would come with a high bandwidth utilization cost.

Each DW1000 UWB's packet contains a long preamble, a long start frame delimeter (SFD), and slow pulse repetition rate to improve the accuracy of the packet reception's timestamp estimation. DecaWave recommends two settings for ranging [23]:

- **Long range**: This mode achieves maximum range of as long as 250 m. The data rate is set to 110 kbps with a 1024 symbol-length preamble and a 64 byte SFD. However, this setting results in a 26 ms frame duration.

- **Short duration**: In this mode, the data rate is set to 6.81 Mbps, preamble length

to 128 symbols, and SFD to 8 symbols. The frame length is reduced to 2.25 ms. However, the range reduces to 60 m and the accuracy of the received timestamps is reduced.

For the long range setting, due to the 26 ms packet duration, we can only achieve a 3.5 Hz update rate for eight tags and four anchors in a real-time locating system (RTLS). For the short duration setting, the update rate increases to 10 Hz [23]. In addition to ranging, the UWB channel is used to transfer timestamp data. The final message containing all of the timestamping information, a total of 44 bytes when there are four anchors, has a 4.929 ms or 0.214 ms duration at 110 kbps or 6.81 Mbps, respectively. The preample and SFD are extraneous for this final message since its timestamp is not utilized for synchronization. The duration of these UWB packets is the limiting factor in increasing the number of tags and anchors in the network.

7.6.1 Conventional

A node (node 1) would require five messages to estimate its offset and skew in respect to the reference node (node 2) through the most simple messaging exchange method [5], which we denote as the conventional method in Figure 7.11. This method provides each node with two timestamps, required for clock offset and skew estimation. Each message carries the previous timestamp; therefore, a "Final" message is required to transmit the final timestamp to node 1. In some radios, it is possible to forward the timestamp in the last response from the reference node or precalculated transmission timestamp prior to the event. However, this precalculated timestamp adds uncertainty since it is a prediction.

The computation for skew Δ_f and time-of-flight ToF in the conventional method is:

$$\begin{aligned} \Delta_f &= \frac{T_{RX_2}[i] - T_{RX_2}[i-1]}{T_{TX_1}[i] - T_{TX_1}[i-1]} \\ ToF &= \frac{T_{RX_1}[i] - T_{TX_1}[i] - T_{TX_2}[i] + T_{RX_2}[i]}{2}. \end{aligned} \tag{7.4}$$

The range estimate between nodes 1 and 2 is simply ToF multiplied by the speed of light in air, hence, we refer to both range and ToF estimation interchangeably. Assuming TX_p and RX_p, radio's processing delays, remains constant, the global time can represented as follows:

$$T_{RX_2}[i] = T_{TX_1}[i] + TX_p + ToF + RX_p - \epsilon \tag{7.5}$$

$$T_{RX_1}[i] = T_{TX_2}[i] + TX_p + ToF + RX_p + \epsilon, \tag{7.6}$$

where ϵ is the clock offset between the two nodes. Yet we are still assuming the clocks are highly stable between message exchanges. As a result, the remaining error would base on the timestamping accuracy.

7.6.2 PolyPoint

The conventional ToF messaging scheme requires five UWB messages to determine clock offset, skew, and ToF estimates. PolyPoint cleverly optimizes packet counts as shown in the middle of the Figure 7.11 by transmitting a "REF" message immediately followed by a "POLL" message. Since the two messages are identical, node 2 may immediately compute clock skew, which it sends back to node 1. The skew and ToF are given by:

$$\begin{aligned} \Delta_f &= \frac{T_{TX_1}[i] - T_{TX_1}[i-1]}{T_{RX_2}[i] - T_{RX_2}[i-1]} \\ ToF &= \frac{T_{RX_1}[i] - T_{TX_1}[i] - \Delta_f(T_{TX_2}[i] - T_{RX_2}[i])}{2}. \end{aligned} \tag{7.7}$$

Compared to (7.4), the protocol in (7.7) requires floating point multiplication to compute ToF, and as such, requires a more capable processing unit. However, the total number of UWB messages is reduced from five to four, a 20% reduction.

7.6.3 EffToF

EffToF is an efficient time-of-flight estimation protocol. In order to reduce the unnecessary UWB channel utilization, EffToF leverages the benefit from having synchronized clocks provided by RFS and secondary narrow-band transceiver, TI CC1200. The CC1200 is, however, not capable of producing high-resolution timestamps, thus, it is still beneficial to use DW1000 packet timestamps. Figure 7.12 shows the proposed setup. Again, we will use the adaptive clock and digital routing to connect up the subsystems of interest and distribute 38.4 MHz LO to drive both of the subsystems (CC1200 & DW1000).

By utilizing the frequency synchronized clock, we can reduce the messages to just those required for time-of-flight estimation. As a result, we are introducing a protocol for efficient DecaWave time synchronization and two-way ranging scheme, as shown in

Figure 7.13. The blue boxes denote transmissions with the DecaWave DW1000 UWB transceiver while the red boxes denote transmissions of the CC1200 narrowband transceiver. In "Frequency & Time Synchronization" operation, the reference node would transmit a pure tone carrier wave for devices that wish to frequency synchronize, as shown in Figure 7.13. Those devices would execute the RFS procedure to correct their frequency offset. Then the 2-way ranging/ToF operation would allow the node to compute its offset and the ToF. For typical applications and environments, we expect that frequency synchronization would be rare compared to ToF estimation; hence, ToF messaging can be simplified to the 2-way ranging operation, that is, the POLL and RESP messages from the UWB and the FINAL message from the CC1200. The ToF and time offset estimate are given as:

$$ToF = \frac{T_{TX_1}[i] - T_{RX_1}[i] - T_{TX_2}[i] + T_{RX_2}[i]}{2} \tag{7.8}$$

$T_{RX_2}[i]$ and $\delta = T_{TX_2}[i] - T_{RX_2[i]}$ will be sent to the node through CC1200 radio message in order to minimize the use of UWB channel like mentioned previously.

In Table 7.1, we show the content of what we think should be in the message (function code, node identification and range set number) to help associate timestamps to round of IEEE 802.15.4 standard exchange. We believe these are the least amount of data for the message exchange in order to compute ToF, time offset (ϵ), and Δ_f. For the sake of simplicity, we have assumed that the target application requires a long range two-way message exchanges. In the following analysis, the DW1000 will be configured to 110 kbps, 1024 preamble symbols, and 64 SFD symbols as the optimal configuration for the long range operation.

With the purpose of observing the scalability of EffToF, we provide in Table 7.2 the total utilization of the UWB channel in μs as a function of A, the number of devices to be time-synchronized. We include only the minimal duration for message exchanges. We do not include the guard time and any additional delays for robust operations. Therefore, the duration is in milliseconds assuming the network contains 5 devices. As stated above, the EffToF's final message is assumed to transmitted with CC1200 on the narrowband channel with the 1.25 Mbps data rate. In this particular example application, we are able to achieve 69.87% less air time on the UWB channel than the conventional approach and 59.23% less compared to our best known method.

7.6.4 Performance Evaluation

We will quantify the effect of frequency synchronization on ToF estimations. Finally, we evaluate EffToF via four experiments.

7.6.4.1 Frequency Offset versus ToF

We fabricate two DecaWave DW1000 capes, as shown in Figure 7.1, to allow the use of an external clock source. In this experiment, instead of using the local oscillator on each of the Chronos, we use an arbitrary waveform generator, Rigol DG4102, to generate 38.4 MHz sinusoidal signal on each of the channel, which is connected to each of the cape as the radio reference clock signal. We start each set of measurement by aligning the phase and validating such operation with an oscilloscope. The ToF are measured following the two-way range message exchange shown in Figure 7.13.

The frequency of the radio reference clock signal is increased by 0.1 to 50 Hz. At each offset, the ToF between the two UWB devices is measured. The results are shown in Figure 7.14. As the frequency offset between the two increases beyond 1 Hz, the ToF (and thus ranging) error increases significantly.

7.6.4.2 EffToF Two-way Ranging

We use two identical Chronos, Beaglebone Black and DecaWave cape setup as shown in Figure 7.1. The UWB antenna, however, is mounted vertically from the board rather than paralleling to the platform like shown in the picture. The devices are calibrated for the correct antenna delay following the recommended procedures [24]. These two nodes are each placed on a polymer tool cart, and the carts are placed at a distance apart in a large office building. The true distance is measured with a Fluke 414D laser range finder, which has a 50 m maximum range. In these experiments, we switch off between the two devices as reference to remove any device bias. On each node, the FPGA on the Chronos synthesizes the LO frequency, and routes the required LO signal to both of the CC1200 subsystems and the external DecaWave DW1000 cape. The Beaglebone Black synchronizes the local oscillator with RFS as described in Section 7.5. The communicatation for ToF follows EffToF protocol, as described in Section 7.6. A ToF measurement is taken about 2 seconds apart. We records 200 measurements at each true range. When the true

range is greater than 50 m, or the line-of-sight (LOS) is blocked, we survey the area and obstructing walls to measure the true distance. The LOS experiments are conducted in different hallways of an office building and inside a partitioned office area. The NLOS tests are conducted through the walls of multiple neighboring offices. Surveying is used to come up with a true distance; hence, cm level errors will be introduced into the true distance.

Through all of these experiments, we achieve RMSE of 17.1 cm for range measurement shown in Figure 7.15 and Figure 7.16. Note, when the actual range is > 35 m, we observe a somewhat higher standard deviation. This may be due to the surveying method used to determine true distance between the antennas. Moreover, although the experiments were conducted during hours when few people are present in the building, during the 60 m range experiment, a person walked along the LOS the measurements. This was the only data point with range standard deviation more than 0.13 cm.

SurePoint, an improved version of Polypoint, reports a 8 cm median error in range estimation [21]. EffToF were able to achieve the same level of performance as other ranging solution, a 17.5 cm median error with a 3.15 cm standard deviation, across all the ranges in several different environments. Even though the standard deviation of the measurements are low in the result, we still have a positive bias in all of your range estimates. This could easily be compensated through calibration.

7.7 Conclusions

In this paper, we introduce Stitch, a unique architecture that allows clock unification and adaptive clock distribution across subsystems. We present the Chronos board, a Stitch-based platform for the testing of a wide range of time- and frequency-synchronized wireless embedded systems. We present RFS and show that, using low-cost narrowband transceivers that two devices can achieve LO frequency synchronization within 3.5 ppb. With such highly synchronous LOs, it is possible to design UWB-based ranging protocols that dramatically reduce their utilization of the UWB channel. We develop and demonstrate the ability to perform efficient time synchronization and two way time-of-flight ranging, in a protocol we call EffToF. EffToF achieves accurate ranging, with 17.1 cm RMSE at indoor ranges up to 75 m, using 59% less of the UWB channel than the state-of-the-art

system. By reducing UWB channel utilization, we enable UWB-based systems to increase the number of time-synchronized devices, or their update rate, or both.

RFS, as described here, uses an unmodulated carrier wave for synchronization. Future work could perform synchronization using the IQ samples from a modulated signal, which would allow frequency sync during data transmission, and also potentially allow the receiver to detect and avoid updating the LO when interference power is high. Future work could implement RFS on the dedicated logic on the FPGA, which would remove the need to use PRU and the main processing core for this task. In other words, the main processors would be available for other tasks and/or sleep to conserve energy.

More broadly, this paper contributes to our ability to achieve frequency synchronization on the order of a few parts per billion across a network of devices using low-cost, low-power, and bandwidth efficient methods. We suggest Stitch and RFS be tested in applications such as distributed beamforming and MIMO, improving the robustness of communications infrastructure to GPS jamming, and enabling new highly-synchronous wireless sensing and actuation systems.

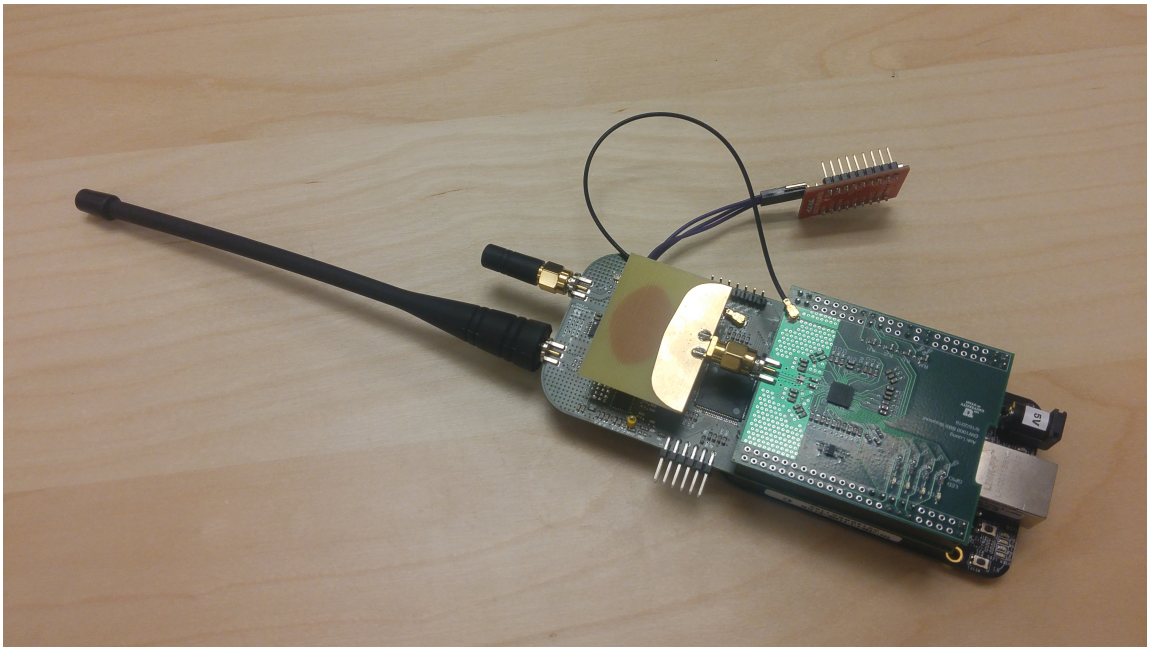


Figure 7.1. EffToF node: Beaglebone Black (bottom), Chronos board (middle), & DW1000 cape (top).

Table 7.1. Message exchange format and duration.

System	Msg Type	Bytes	Duration (μs)	Channel	Contents
Conventional	ALL	21	3026	UWB	fn, node#, Ttrx[i], range#
PolyPoint	REF / POLL	13	2501	UWB	fn, range#
	RESP	29	3551	UWB	fn, node#, Trx[i-1], Trx[i], range#
	FINAL	21	3026	UWB	fn, node#, Ttx[i], range#
EffToF	POLL	13	2501	UWB	fn, range#
	RESP	14	2566	UWB	fn, node#, range#
	FINAL	29	186	NB	fn, node#, Trx[i], Ttx[i], range#

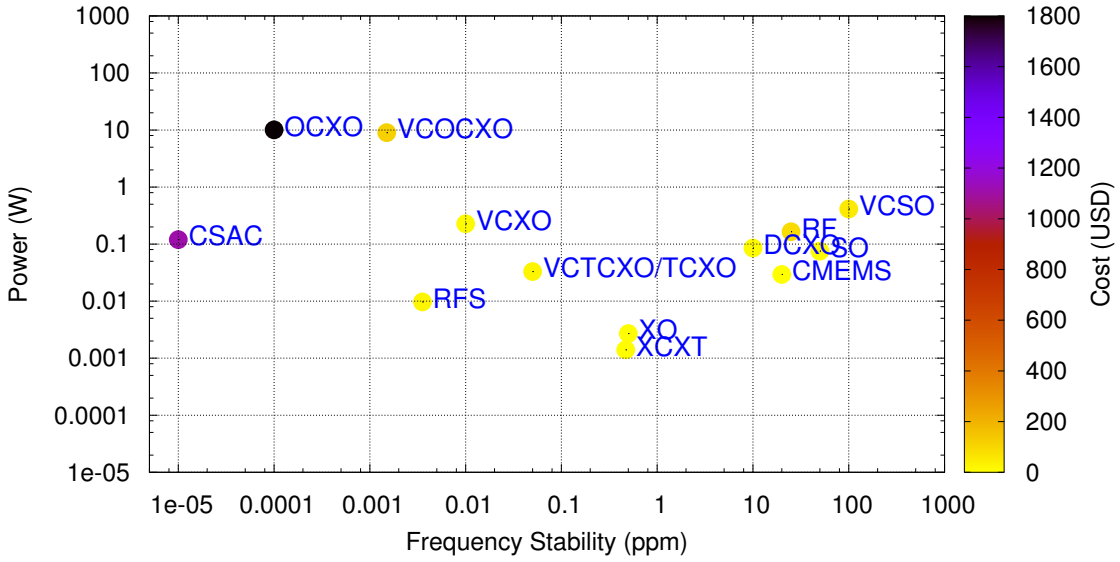


Figure 7.2. RFS provides a uniquely low-power, low-cost (shown for quantity = 1), and highly stable oscillator, compared to other commercially available oscillators.

Table 7.2. UWB ToF message duration for network of 5 nodes.

Protocol	Duration (A, ms)	Duration (A = 5, ms)
Conventional	$3.026 \times (3A - 1)$	42.364
PolyPoint	$2.501 \times 2 + (3.551 + 3.026) \times (A - 1)$	31.310
RFS	$2.501 + 2.566 \times (A - 1)$	12.765

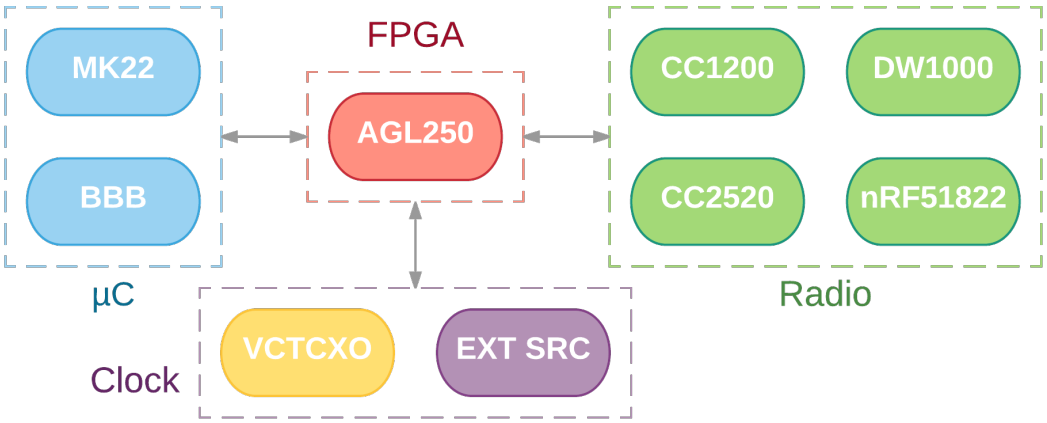


Figure 7.3. Chronos: Adaptable research platform, a Beaglebone cape with a Freescale MK22, Microsemi IGLOO FPGA, multiple radios, 1 ppm 40 or 38.4 MHz VCTCXO, and clock I/O.

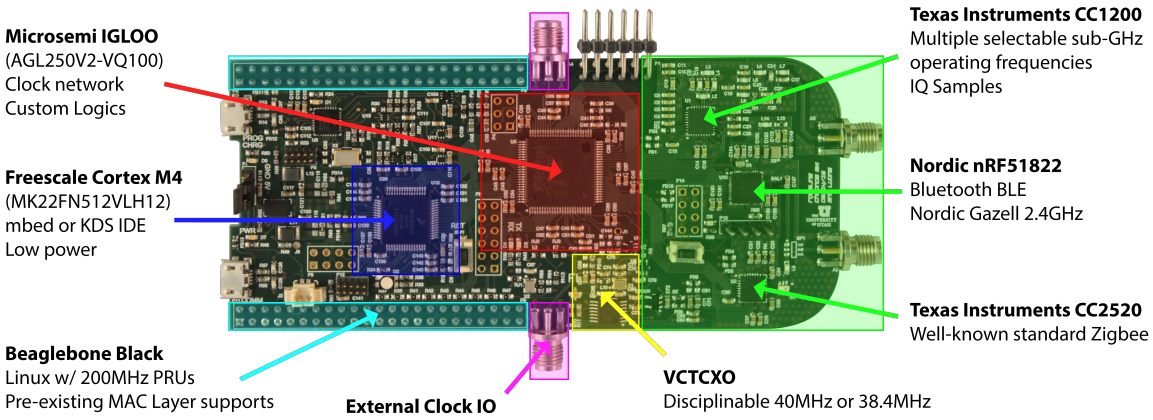


Figure 7.4. Chronos: on-board μP and VCTCXO, optional Beaglebone Black and clock I/O, and radios, adaptively wired via FPGA.

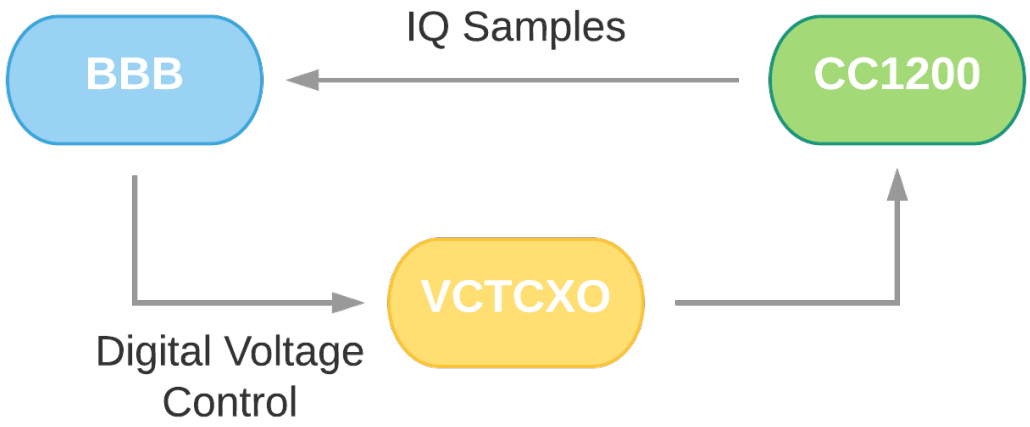


Figure 7.5. A Beaglebone Black PRU collects complex baseband samples from the CC1200, computes the frequency offset and corrects the shared VCTCXO. The algorithm iterates until the frequency difference between two nodes is zero.

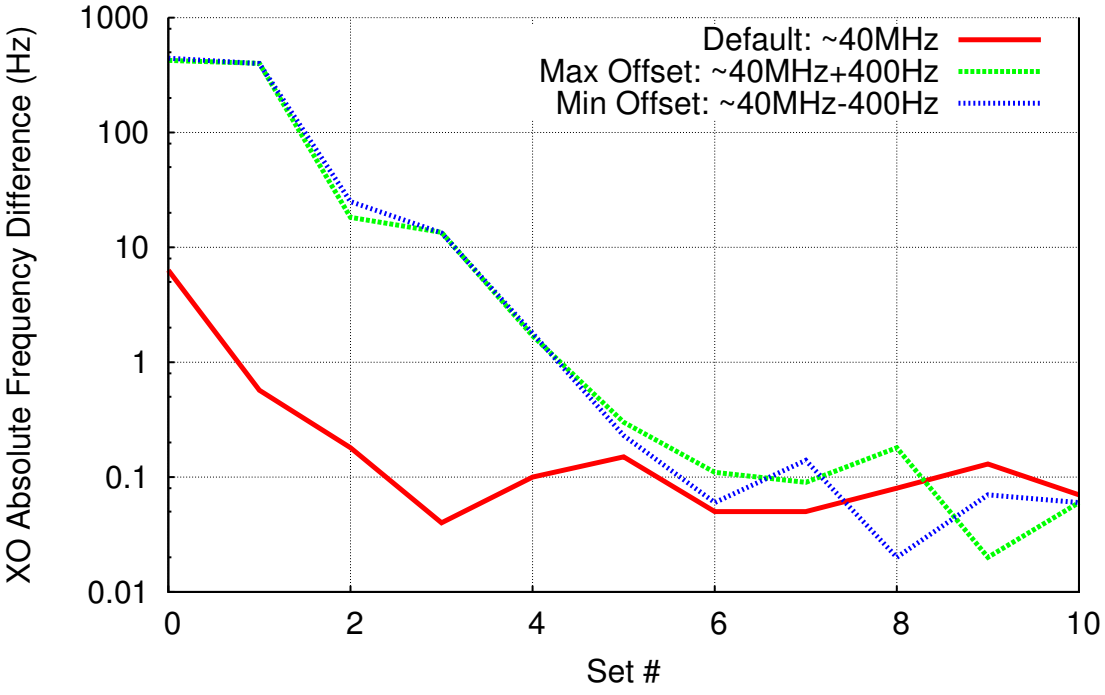


Figure 7.6. Frequency difference vs. iteration and LO starting frequency. The naive algorithm ($N = 1000$) synchs the 40 MHz LO within ± 0.1 Hz in ≤ 6 iterations.

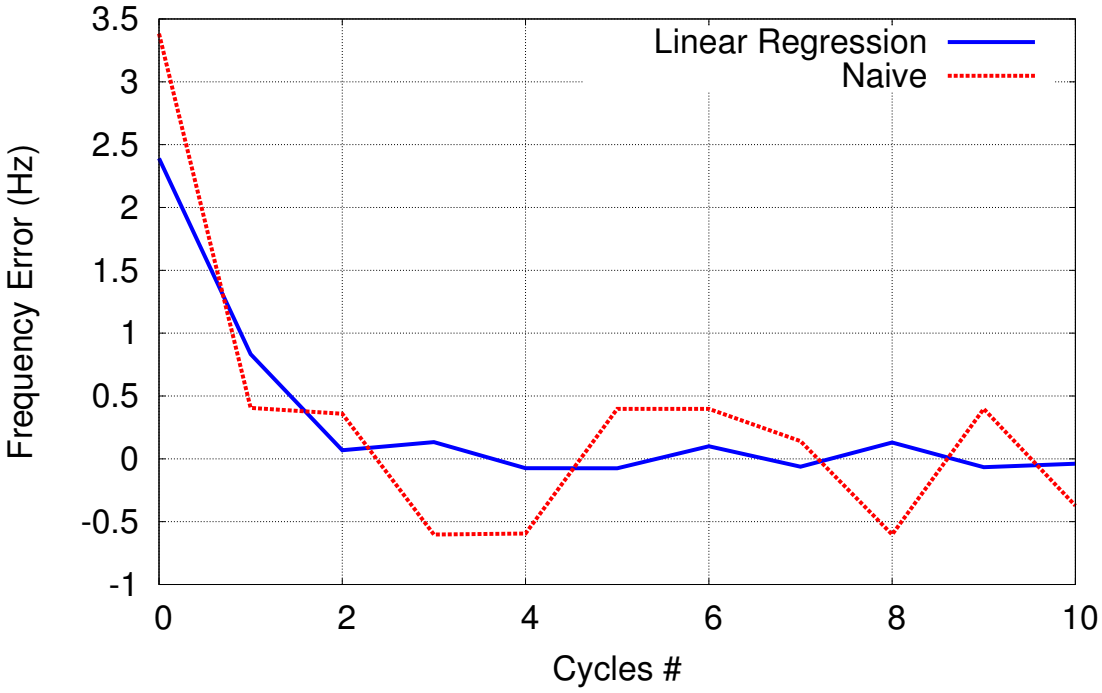


Figure 7.7. Naive vs. linear regression algorithm performace in RFS.

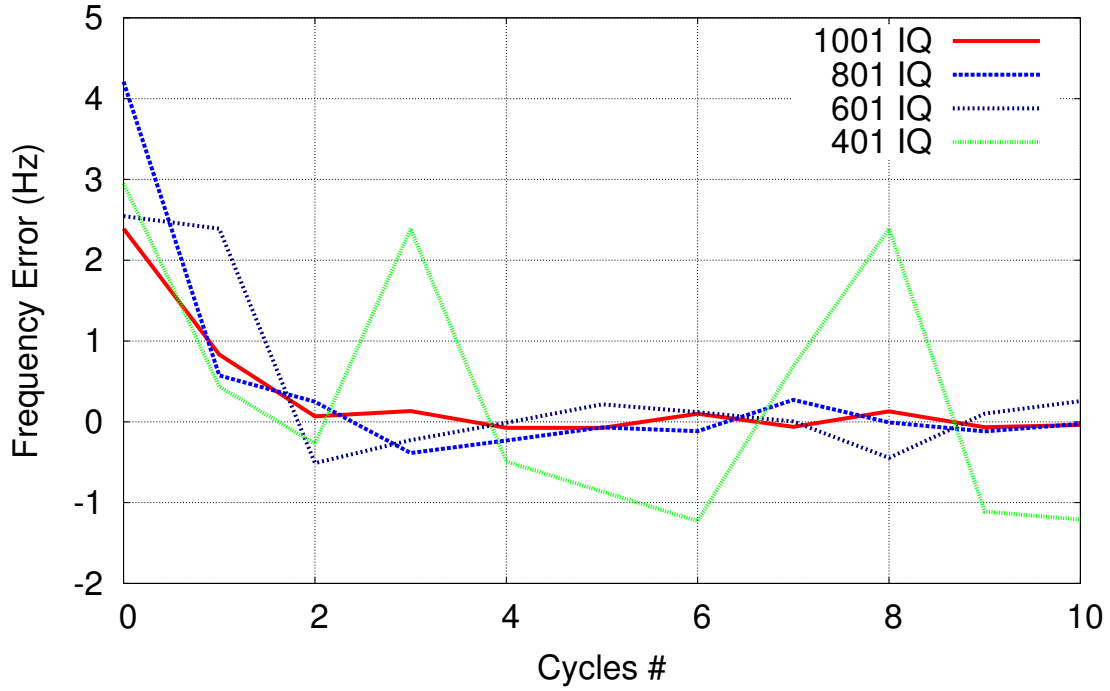


Figure 7.8. Frequency synchronization accuracy vs. number of samples N, using the linear regression estimator.

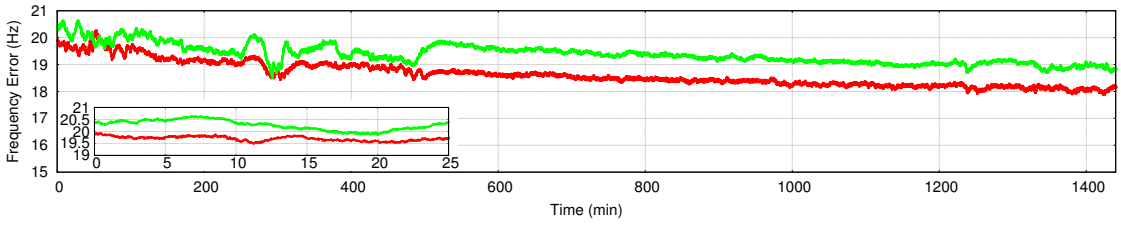


Figure 7.9. Absolute frequency difference of two uncompensated LOs over a 24-hour period. The frequency difference between the two devices is mostly less than 1 Hz in the first 25 minutes (sub-figure). After the minute 450, the frequency offset increases but stays relatively constant.

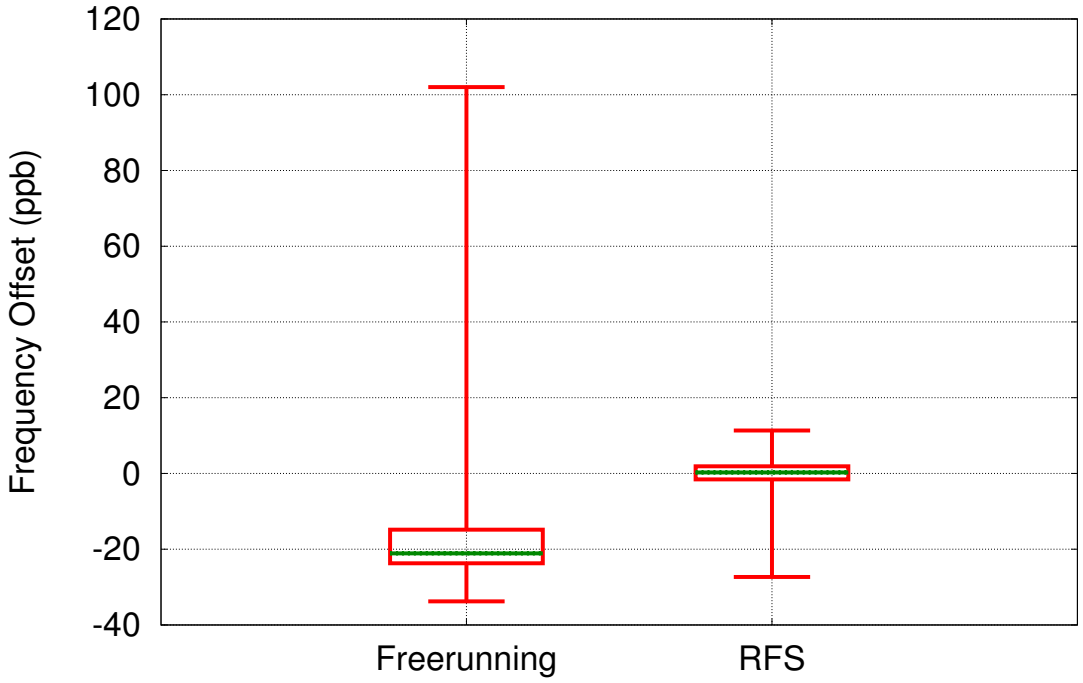


Figure 7.10. Box plot of LO frequency error over 24 hours, freerunning vs. RFS. We achieve an RMSE of 0.136 Hz with RFS once every 20 minutes while the freerunning method has RMSE of 0.773 Hz.

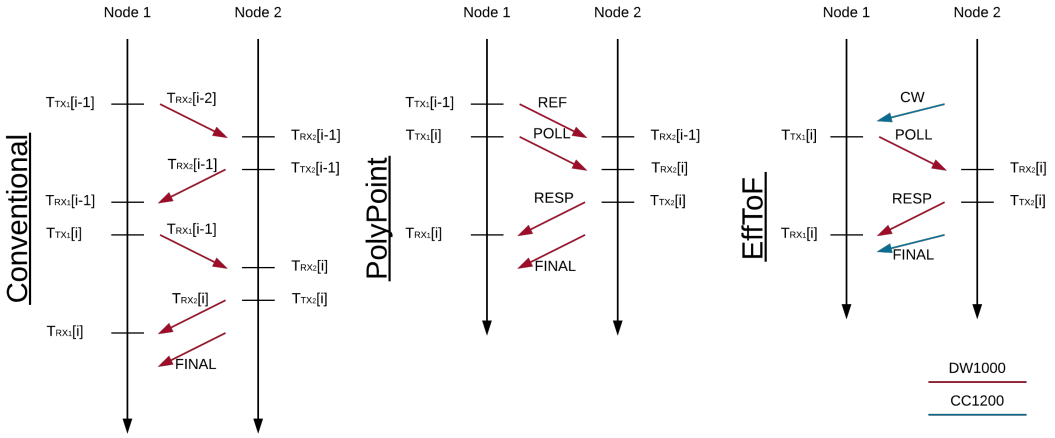


Figure 7.11. Timestamp protocols: (Left) conventional message exchange; (Middle) PolyPoint; (Right) proposed EffToF scheme.

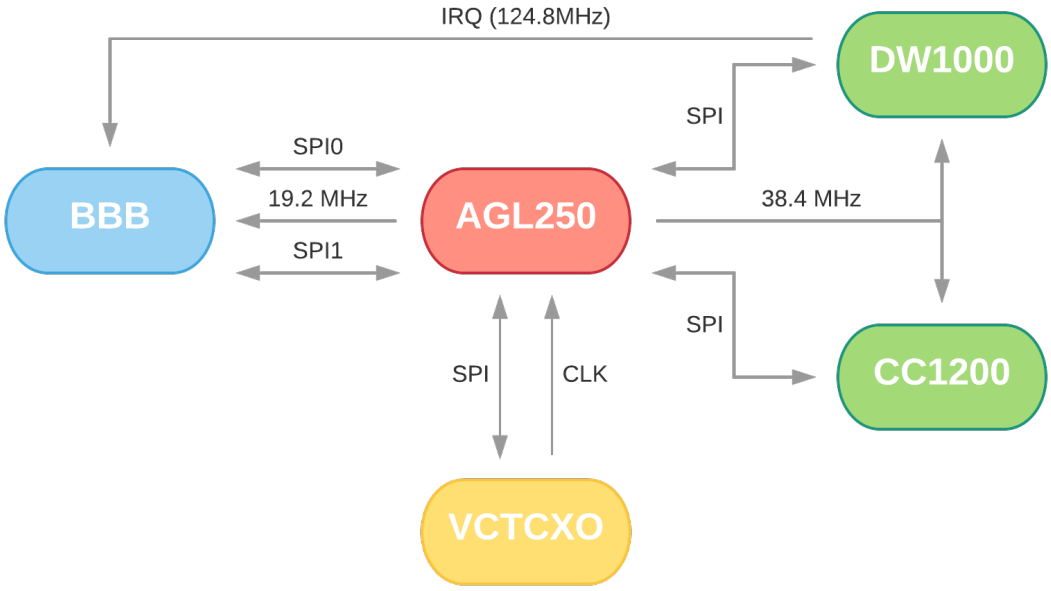


Figure 7.12. Proposed circuitry and submodules for accurate time synchronization.

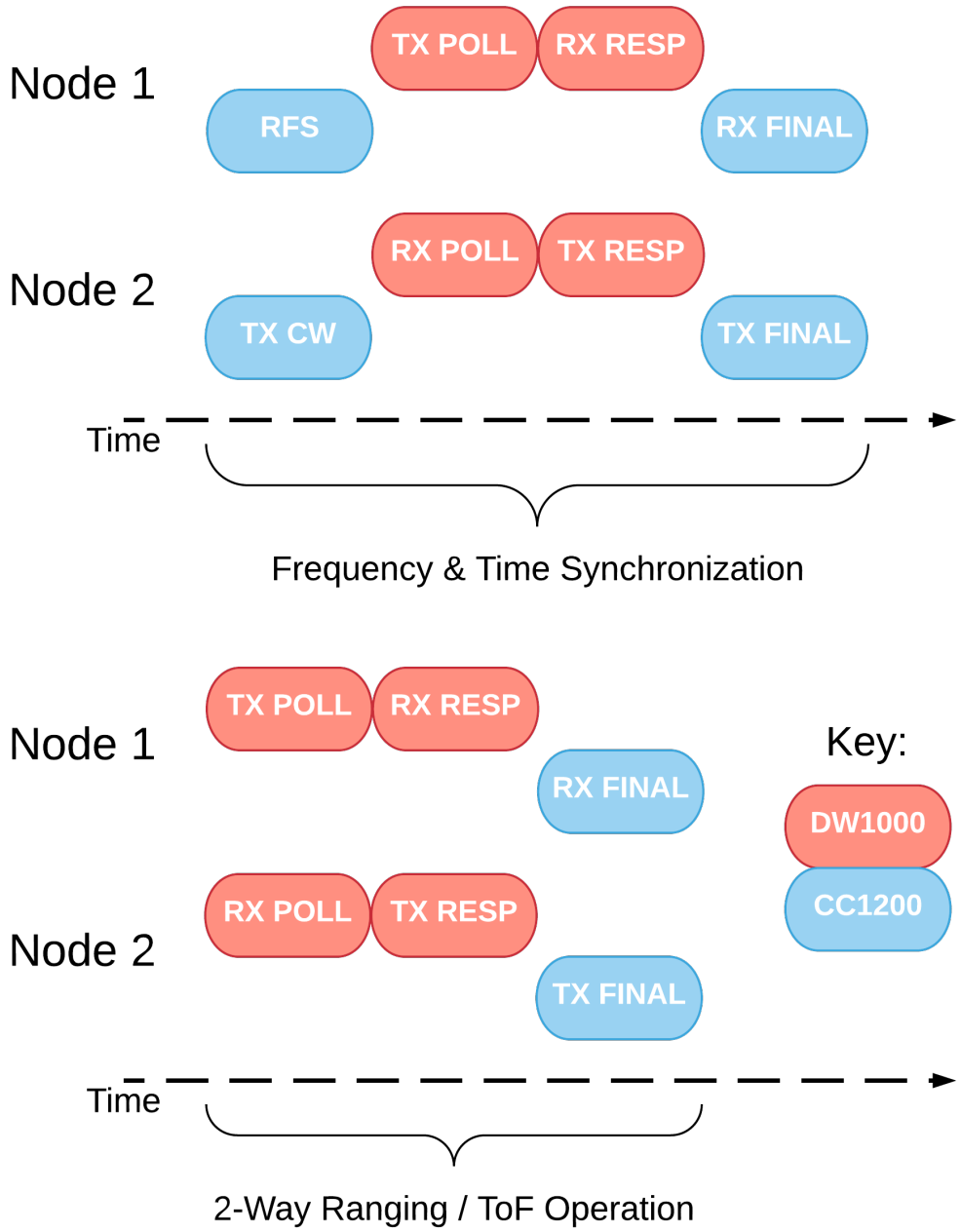


Figure 7.13. TDMA Scheme to achieve a single DW1000 message ranging or time synchronization.

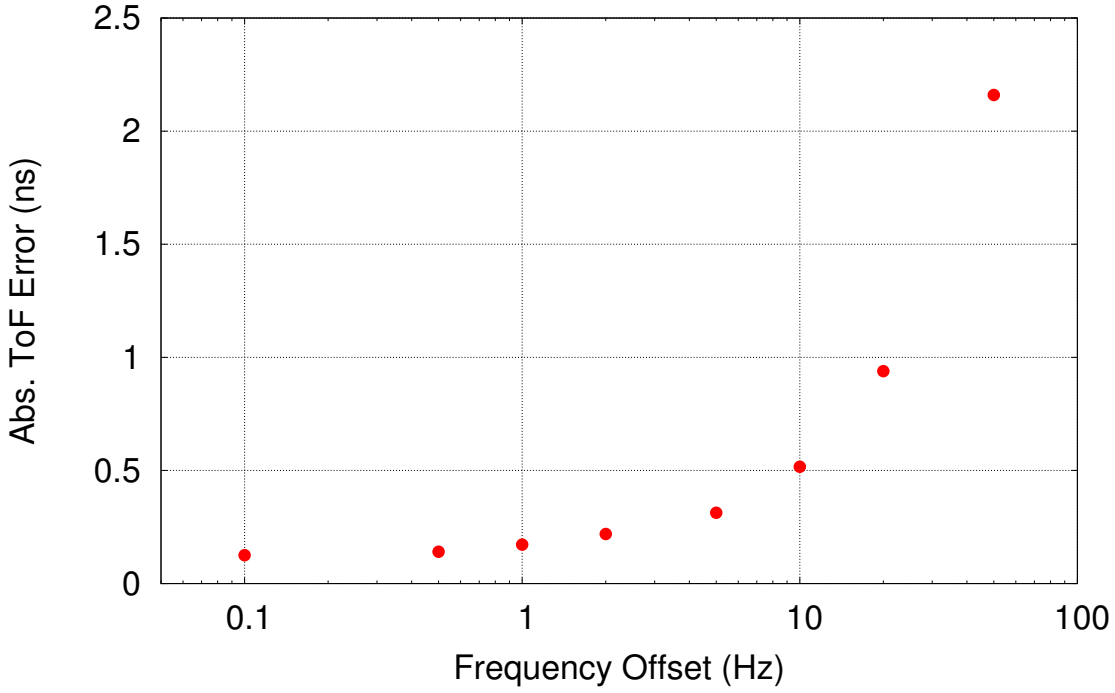


Figure 7.14. Frequency offset versus ToF. The absolute ToF error increases as a function of frequency offset between the two LOs.

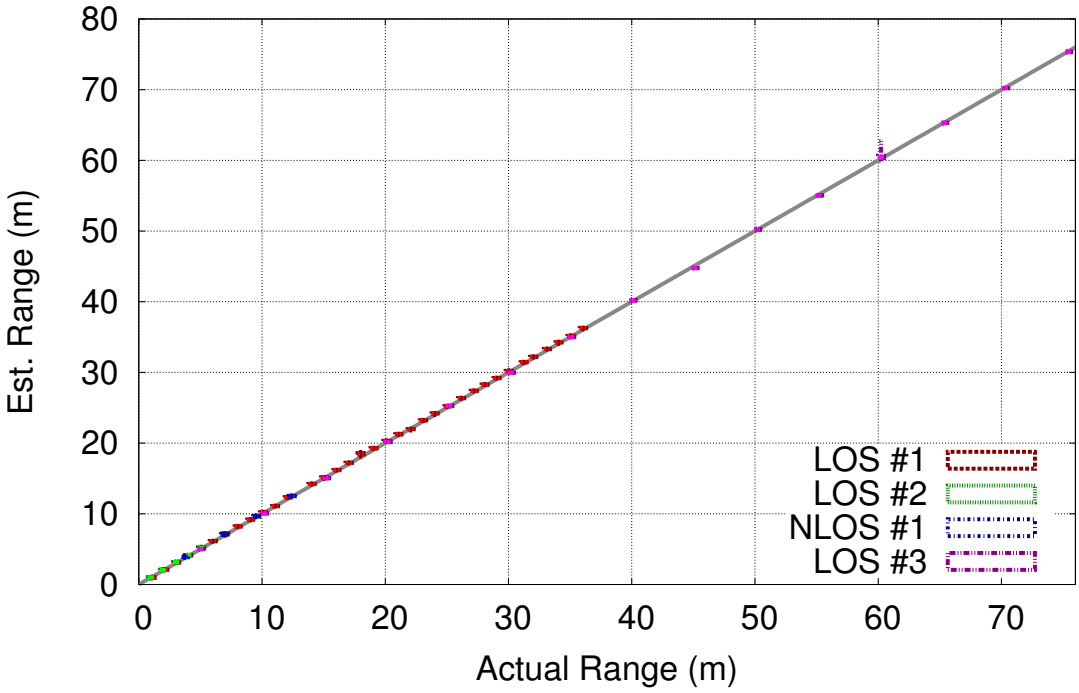


Figure 7.15. Box plots of estimated vs. actual range over LOS and NLOS experiments.

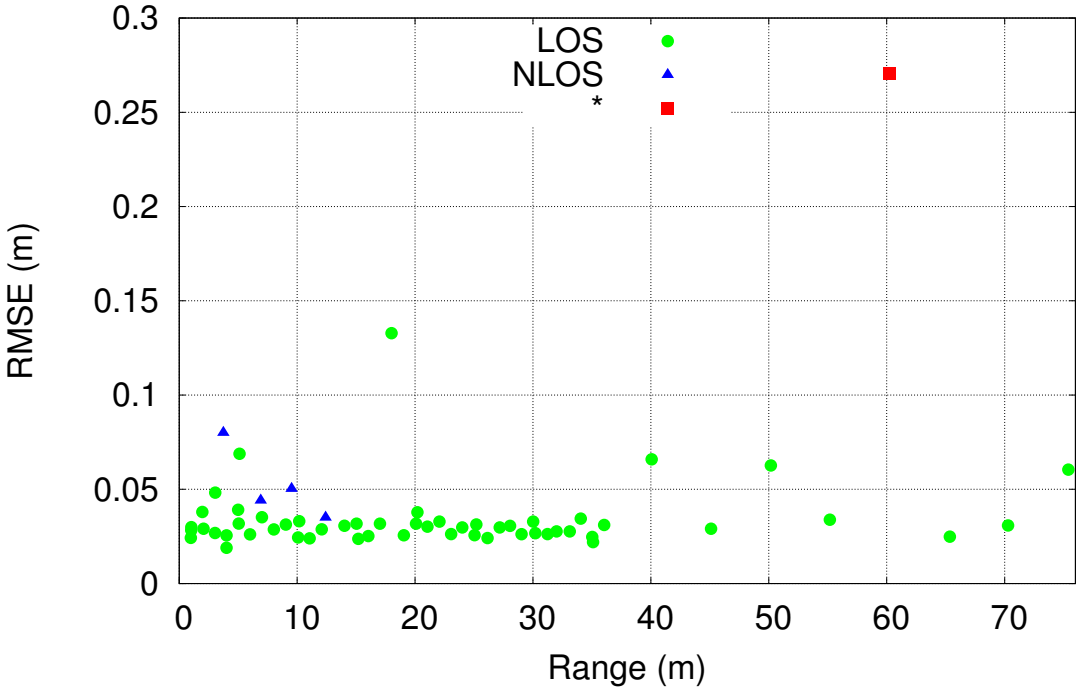


Figure 7.16. RMSE and standard deviation of LOS and NLOS ranging experiments. In one experiment (*), a person walked along the link line during the ToF measurement.

7.8 References

[1] Dimitrios Lymberopoulos, Romit Roy Choudhury, Jie Liu, Souvik Sen, Xue Yang, and Vlado Handzinski, "Microsoft indoor localization competition: Experiences and lessons learned," *SIGMOBILE Mobile Computation and Communication Review (MC2R)*, October 2014.

[2] "Electronic code of federal regulations," 2017, http://www.ecfr.gov (Title 47, Chapter I, Subchapter A, Part 15, Subpart F).

[3] Benjamin Kempke, Pat Pannuto, and Prabal Dutta, "Polypoint: Guiding indoor quadrotors with ultra-wideband localization," in *Proceedings of the 2nd International Workshop on Hot Topics in Wireless*. ACM, 2015, pp. 16–20.

[4] Jeremy Elson, Lewis Girod, and Deborah Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002.

[5] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 138–149.

[6] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 39–49.

[7] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh, "Efficient network flooding and time synchronization with glossy," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*. IEEE, 2011, pp. 73–84.

[8] Christoph Lenzen, Philipp Sommer, and Roger Wattenhofer, "Pulsesync: An efficient and scalable clock synchronization protocol," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 3, pp. 717–727, 2015.

[9] Michael A Lombardi, "The use of gps disciplined oscillators as primary frequency standards for calibration and metrology laboratories," *NCSLI Measure*, vol. 3, no. 3, pp. 56–65, 2008.

[10] "Abracon," AOCJY6 Series Datasheet.

[11] "Microsemi," Quantum SA.45s CSAC Datasheet.

[12] Thomas Schmid, *Time in wireless embedded systems*, Ph.D. thesis, University of California Los Angeles, 2009.

[13] Adwait Dongare and Anthony Rowe, "Propagation-aware time synchronization for indoor applications: Demo abstract," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 292–293.

[14] Maxim Buevich, Niranjini Rajagopal, and Anthony Rowe, "Hardware assisted clock synchronization for real-time sensor networks," in *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*. IEEE, 2013, pp. 268–277.

[15] David Jenn, Yong Loke, Tong Chin Hong Matthew, Yeo Eng Choon, Ong Chin Siang, and Yeo Siew Yam, "Distributed phased arrays and wireless beamforming networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 4, pp. 283–302, 2009.

[16] David Jenn, Yong Loke, Matthew Tong, Eng Choon Yeo, and Robert Broadston, "Distributed phased arrays with wireless beamforming," in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*. IEEE, 2007, pp. 948–952.

[17] Ibrahim Kocaman, *Distributed beamforming in a swarm UAV network*, Ph.D. thesis, Monterey California. Naval Postgraduate School, 2008.

[18] Yoke Chuang Yong, *Receive channel architecture and transmission system for digital array radar*, Ph.D. thesis, Monterey, California. Naval Postgraduate School, 2005.

[19] Michael Rice, *Digital Communications: a Discrete-Time Approach*, Pearson Prentice Hall, 2009.

[20] DecaWave, ," ScenSor DW1000. http://www.decawave.com/.

[21] Benjamin Kempke, Pat Pannuto, Bradford Campbell, and Prabal Dutta, "SurePoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization," in *Proceedings of the 14th ACM Conference on Embedded Networked Sensor Systems*, November 2016, SenSys'16.

[22] Thomas Schmid, Prabal Dutta, and Mani B Srivastava, "High-resolution, low-power time synchronization an oxymoron no more," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2010, pp. 151–161.

[23] DecaWave, ," DecaRangeRTLS ARM Source Code Guide.

[24] DecaWave, ," Antenna delay calibration of DW1000-based products and systems.

CHAPTER 8

CONCLUSION

In search of a solution for search and rescue applications, this dissertation explores multiple aspects for wireless sensing networks. First, the design of systems that use RSSI is evaluated for device-free localization, device-based localization, and breathing monitoring applications. The use of standard RSSI imposes multiple limitations on the accuracy and reliability of such systems, but higher accuracy is available only with very high system costs, both in bandwidth and device costs. Second, this dissertation presents the design and evaluation of novel wireless embedded systems that are able to provide more fine-grained radio signal measurements to the application. The effect of increasing the resolution of a RSSI beyond the typical 1 dB step size, which is the current standard, is studied with a sub-dB system based on this novel capability through two example applications, breathing monitoring and gesture recognition. Third, Chronos implemented the Stitch architecture, which allows the testing of a wide range of time- and frequency-synchronized wireless embedded systems. RFS is implemented to use low-cost narrowband transceivers to achieve LO frequency synchronization of 3.5 ppb between the two independent devices. With such highly synchronous LOs, EffToF is designed as UWB-based ranging protocols that dramatically reduce their utilization of the UWB channel. EffToF achieves accurate ranging, with 17.1 cm RMSE at indoor ranges up to 75 m, using 59% less of the UWB channel than the state-of-the-art system. By reducing UWB channel utilization, EffToF enables UWB-based systems to increase the number of time-synchronized devices, their update rate, or both. This dissertation, in short, addresses fundamental challenges in the aspects of designing, managing, and deploying of scalable wireless sensor networks with WRENSys, Dial-It-In, and boundary crossing system. While building the first real-time breathing monitoring system, we came to realize the limits of the RSSI-based sensing. With conventional RSSI-based sensing, we were unable to detect all of the

small changes in the environment. Thus, we were pushed towards other information-rich solutions. As a result, sub-dB was developed as a new sensing capability to push the boundary, to improve the current and enable new RSSI-based sensing applications (i.e., human activity monitoring, heart rate monitoring, etc...). This dissertation also discusses a complete system for efficient synchronization and localization that would enable a new wave of future applications.

8.1 Lesson Learned and Future Work

In WRENSys, a large-scale system is achievable with correct architecture and design. As a team, we designed and assembled the prototypes of the system in-house by hand, thus resulting in a much faster turnaround time. The logistics of manufacturing the hardware for deployment and conducting multiple deployments at the same time became a challenge. With five members, we were able to successfully build and deploy thousands of WRENs within 6 months. However, the system is still imperfect. Due to the deployment environment, devices are often plugged in rapidly, resulting in bad connections with the base station. This often causes the node to not recharge or program properly. In future work, I would highly recommend fellow researchers to look at other solutions for devices' connections. Moreover, the mass programming bus uses I2C, which has an inherit problem as the external bus. The protocol was designed as a communication method between chips, thus would not have to face signal integrity challenge in WRENSys. A differential interface and protocol would greatly improve the scalability of mass programming. After the end of the project, Prof. Ross Walker recommended the use of optocouplers on the signal lines of I2C. Unfortunately, we could not evaluate the recommendation in this dissertation. Nowadays, wireless programming, a.k.a. over-the-air (OTA) upgrade, is a built-in feature for SOCs, which could use the same broadcast based principle used in WRENSys for fast programming. However, this option was not available during the time of this project due to hardware limitations, which is mostly due to the fact that BSL Flash does not have the capacity to contain the wireless communication stack.

With Dial-It-In, the nodes are relatively simple to build. However, the servo requires high power consumption. A node could be build with different antenna and switch through RF switcher would be much more power efficient. However, it will significantly

increase the cost of the devices. In addition, the devices might only need to be optimized once in the given deployment environment; thus, for permanent fixture, this solution is rather resourceful.

A boundary crossing detection system brings different sets of challenges: weather protection for the nodes and a sustainable powering solution. In addition, the scale of the experiment is limited by the 2.4 GHz transmission power. Lower frequency or higher transmission power is highly recommended for realistic commercial deployment.

While sub-dB contains all easily obtainable breakout boards, the device is simply a prototype as of yet. Therefore, a custom platform can be built based on hardware shown rather than replication of this hardware for commercial purposes. Through rough estimation, a custom platform with the CC1200 and Cortex-M4 with BLE 5 capabilities would cost less than 20 USD for parts (quantity = 1k units).

Stitch contains the fundamental building blocks for systems requiring an adaptive controllable clock. This dissertation demonstrates Stitch with an FPGA, a custom clock source (VCTCXO and DAC), and TI CC1200; however, the idea could be extended to other hardware such as voltage control oven-compensated crystal oscillator (VCOCXO) with high resolution 32-bits DAC and Atmel RF233. Moreover, Stitch could also apply to other applications even though I only choose to show its advantages in synchronization and localization. One of those applications is a small 10-pin module to replace the GPS-disciplined oscillators used by the USRP to enable applications like distributed MIMO and wireless beamforming.

Chronos is a research platform based on Stitch, which is an architecture to explore time and frequency synchronization. Chronos contains quite an array of subsystems. Initially, I designed the Chronos as a platform for clock-related wireless sensor network research. In other words, I wanted to create a large testbed of many of these platforms. This would allow testing of a variety of characteristics of clocks (jitter, drift, etc...) and their effects on the performance of wireless sensor network. Beyond these uses, Chronos has proven useful in that it also includes features such as external clocks and external digital signals as future proof in case a newer and more desirable clock source becomes available. In other words, a module could be built and attached to Chronos and used in place of the on-board VCTCXO circuitry.

With the two above sample applications, sub-dB and RFS, I was still unable to fully use the Chronos to its entirety. The CC1200 is shown with the ability to export IQ samples and change operating frequency band in software. This means it could be used as a software defined radio in certain applications, i.e., spectrum sensing. However, a new matching network is required for highest RF performance for all of the center frequencies (169 MHz, 434 MHz, and 868 MHz).

RFS, as described, uses an unmodulated carrier wave for synchronization. Frequency synchronization using the IQ samples from a modulated signal would allow frequency sync during data transmission and also potentially allow the receiver to detect and avoid updating the LO when interference power is high. On another note, RFS can be implemented on the Chronos as dedicated logic on the FPGA, which would remove the need to use PRU and the main processing core for this task. In other words, the main processors would be available for other tasks or sleep to conserve energy. Moreover, RFS could potentially be applied to distributed beamforming and MIMO, improving the robustness of communications infrastructure to GPS jamming, and enabling new highly-synchronous wireless sensing and actuation systems. Unfortunately, it was not possible to test out these applications in this dissertation.

EffToF was only evaluated as a two-way ranging setup. Due to the synchronization requirements, it is challenging to build a wireless time difference of arrival (TDOA) system. However, EffToF allows efficient synchronization of the anchors. Thus, the amount beacon tags or localization update rate is significantly increased. For instance, the helmets used by football player could alert them to trigger unconscious muscle contraction right before a collision to prevent concussion. With the standard system, it might not possible to track all the players and provide them with real-time update rate before a collision. However, I believe it is totally obtainable with the use of EffToF.